AN ABSTRACT OF THE THESIS OF

GARY WAYNE McNAIR for the degree of MASTER OF SCIENCE

in Nuclear Engineering presented on July 19, 1977

Title: AN EVALUATION OF FINITE DIFFERENCE METHODS FOR CALCULATING HEAT TRANSFER IN FUEL PINS WITH ECCENTRICALLY PLACED PELLETS

Redacted for Privacy

Abstract approved: K. L. Peddicord

In fuel pin fabrication, there is no assurance that a fuel pellet will be loaded concentrically. Therefore, azimuthal variations exist in the temperature field giving rise to asymmetry in the neutron flux distribution and pin stresses. Calculations must be made to determine the two dimensional temperature profile which may then be used to evaluate the resulting fuel pellet and cladding stresses and deformations. It is most convenient, in these type calculations, to utilize an existing general purpose finite difference code. However, these codes can only treat concentric regions.

Two general approaches have been used to treat the problem. The first is to approximate the outer boundary by a ratchet. This results in an increase in the number of radial node regions required and a subsequent loss of

economy. The second approach utilizes concentric regions and defines a variable conductivity within the gas gap to account for the eccentric condition. This reduces the number of radial node regions and results in savings in both computation time and core space.

The purpose of this paper is to evaluate these two approaches for power reactor fuel pin calculations and extend the methodology to general eccentric calculations.

An Evaluation of Finite Difference Methods for Calculating Heat Transfer in Fuel Pins with Eccentrically Placed Pellets

Gary Wayne McNair

A THESIS

submitted to

Oregon State University

in partial fulfillment of
the requirements for the
degree of

Master of Science

June 1978

APPROVED:

Redacted for Privacy

Assistant Professor of Nuclear Engineering
in charge of major

Redacted for Privacy

Head of Department of Nuclear Engineering

Redacted for Privacy

Dean of Graduate School

Date thesis is presented July 19, 1977

Typed by Deanna L. Cramer for Gary Wayne McNair

ACKNOWLEDGEMENTS

I would like to express my gratitude to all those who have aided me during my graduate studies here at Oregon State.

I wish to thank Dr. K. L. Peddicord for his advice and encouragement which aided in more ways than he will ever know.

I want to thank T. J. Trapp who has aided enormously by providing a computer plot routine for presenting my results. I wish to also thank his wife, Deb E. Trapp, for drawing the graphs that the plot routine wouldn't do.

I would like to thank my parents for their continuing faith and encouragement during the good times as well as the bad.

Last, but certainly not least, I want to express my thanks and my love to my wife, to whom I have dedicated this thesis.

TABLE OF CONTENTS

LIST OF FIGURES

List of Figures -- continued

AN EVALUATION OF FINITE DIFFERENCE METHODS FOR CALCULATING HEAT TRANSFER IN FUEL PINS WITH ECCENTRICALLY PLACED PELLETS

I. INTRODUCTION

One of the foremost concepts of reactor safety is the concept of multiple containment barriers for fission product retention. In a power reactor fuel pin, the cladding serves as the first and most important containment of radioactive fission products. To assure the complete integrity of the fuel pin cladding, it is necessary to have a thorough understanding of the induced stresses within the pin. One source of such stresses is asymmetry effects due to eccentric loading of fuel pellets within the clad.

In fuel pin fabrication there is no assurance that a fuel pellet will be loaded concentrically within the cladding. Therefore, if the pellet is loaded eccentrically, there will exist azimuthal variations in the temperature field giving rise to asymmetry in both the neutron flux distribution and pin stresses. Calculations must be made to determine the two dimensional temperature profile which may then be used to evaluate the resulting fuel pellet and cladding stresses and deformations. The most convenient method of calculation would utilize an existing standard finite difference code; however, these codes are designed to only treat concentric regions.

Two general approaches have been used to model the eccentrically loaded fuel pin within the confines of the finite difference programs. The first is to approximate the outer boundary of the fuel pellet by a ratchet boundary while the second method utilizes a geometrically dependent variable conductivity within the gas gap to account for the eccentric conditions. These two approaches are presented in Chapter II.

The temperature profiles as predicted by the two codes for various cases are analyzed and compared. The results are presented in Chapter III. At the time of writing, no analytical solution exists to the eccentricity problem. Without the availability of such a solution to provide a benchmark for these studies, it is possible to make only relative comparisons.

Chapter IV introduces modifications to the modeled conductivity approach which result in, as shown in Chapter V, a much more favorable comparison between the temperatures prediced by the two numerical models.

Chapter VI briefly reviews the conclusions of this paper.

II. METHODS OF ANALYSIS

Ratchet Boundary Model

The ratchet boundary model utilizes a single coordinate system to describe the fuel pin cladding. This coordinate system is divided into equal angular regions. Within each region, the distance from the origin of the coordinate system to the midpoint of the fuel pellet surface which bounds that region is determined. This distance, obtained for each region, is then used to construct concentric arcs for each region, which, in total, makes up the fuel pellet's ratchet boundary (see Figure 1).

The length of the radius through the midpoint of each angular section is found by utilizing a second coordinate system centered within the fuel pellet (see Figure 2). From a direct observation of the two coordinate systems, the following relationships may be obtained:

$$\rho \cdot \cos \phi - d = R \cdot \cos \theta \qquad 2.1$$

$$\rho \cdot \sin \phi = R \cdot \sin \theta \qquad 2.2$$

where ρ = radius of ratchet boundary segment

R = radius of fuel pellet

d = distance between origins of coordinate systems

Figure 1. Ratchet boundary approximation.

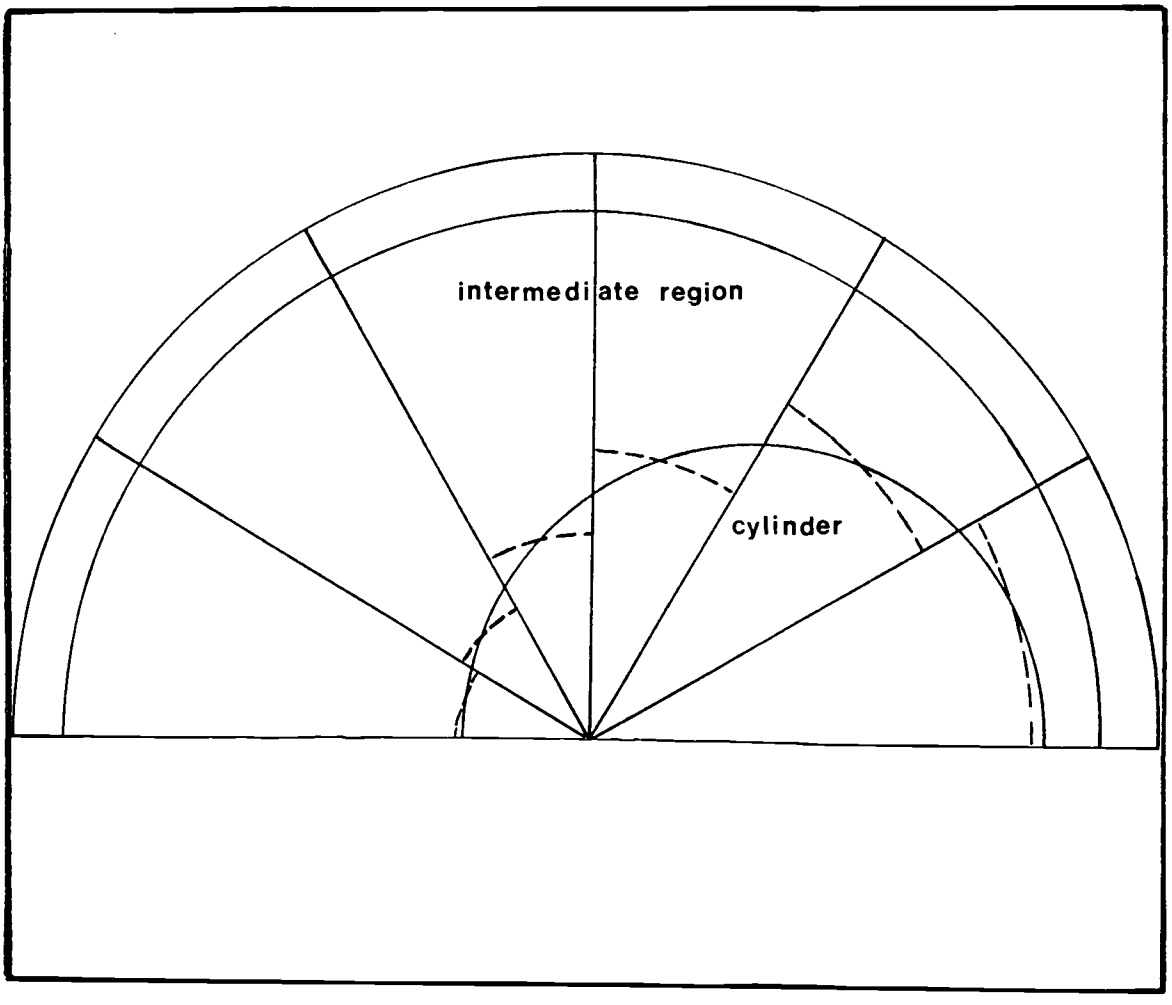
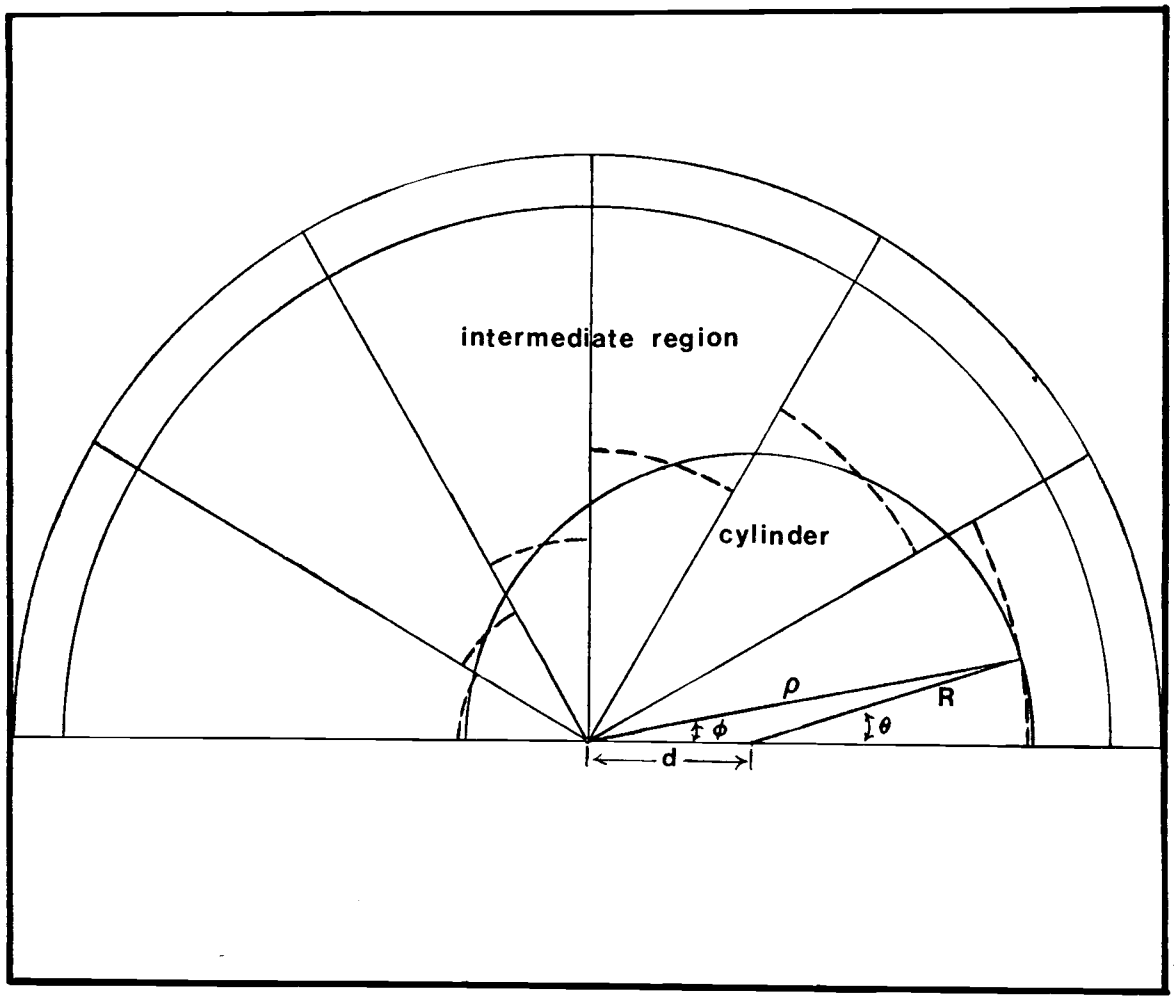


Figure 2. Determination of ratchet boundary radius.



Solving the above equations for ρ as a function of R and ϕ gives the following:

$$\rho = \frac{2d \cos\phi + [(2d \cos \phi)^2 - 4(d^2 - R^2)]^{\frac{1}{2}}}{2} \qquad 2.3$$

which may be simplified to the following form

$$\rho = d \cos \phi + [R^2 - d^2(1 - \cos^2 \phi)]^{\frac{1}{2}} \qquad 2.4$$

The ratchet boundary model, due to its geometry, requires as many radial node regions as there are angular regions to describe the fuel pellet surface alone. In addition to the fuel pellet surface regions, there must be radial node regions to describe the gas gap, the cladding, and the core of the fuel pellet. There are no restrictions placed on the number of angular regions required for the ratchet boundary model.

Due to the single coordinate system centered within the cladding, the ratchet boundary model also requires the temperatures in the fuel pellet to be translated into a coordinate system which has a coordinate axis at the center of the fuel pellet.

A finite difference code, RAT, was written to perform the operations required by the ratchet boundary model. This code, along with a simple code, RFIND, to translate the temperature profile as stated above, are described in Appendix B.

Modeled Conductivity Approach

The modeled conductivity approach utilizes concentric regions and defines a geometrically dependent variable conductivity within the gas gap to account for the eccentric condition (1).

The modeled conductivity program assumes a fuel pin which consists of a fuel pellet of radius R_f located asymmetrically in a can of inner radius R_c and outer radius R_o (see Figure 3). The coordinate system (r,θ) describes the fuel pellet while the system (ρ,ϕ) defines the cladding. The distance between the origins of the respective coordinate systems is given as d (in centimeters). The eccentricity, e, a number between zero and one, indicates the amount which the pellet is shifted within the clad and is related to d by the following equation:

$$d = e(R_c - R_f) \qquad 2.5$$

At the interface between the two coordinate systems, which is arbitrarily taken to be in the gap at some $\rho = R'$, it is required that the component of heat flux normal to this boundary must be continuous. The vector normal to this boundary is defined as $\hat{n}$ and is equal to the unit vector in the ρ-direction (see Figure 4).

To calculate the component of heat flux normal to this boundary, the normal vector must be dotted into the

Figure 3. Modeled conductivity system.

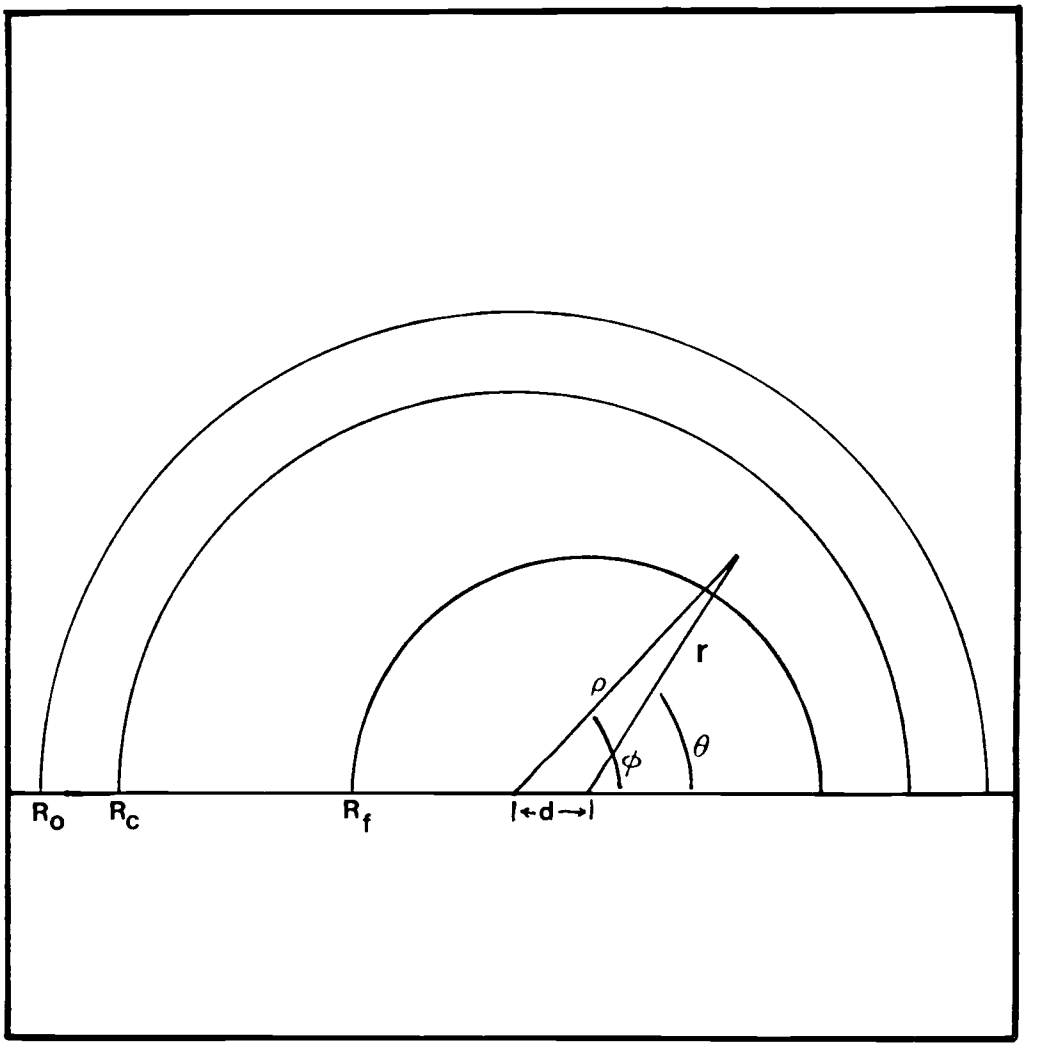
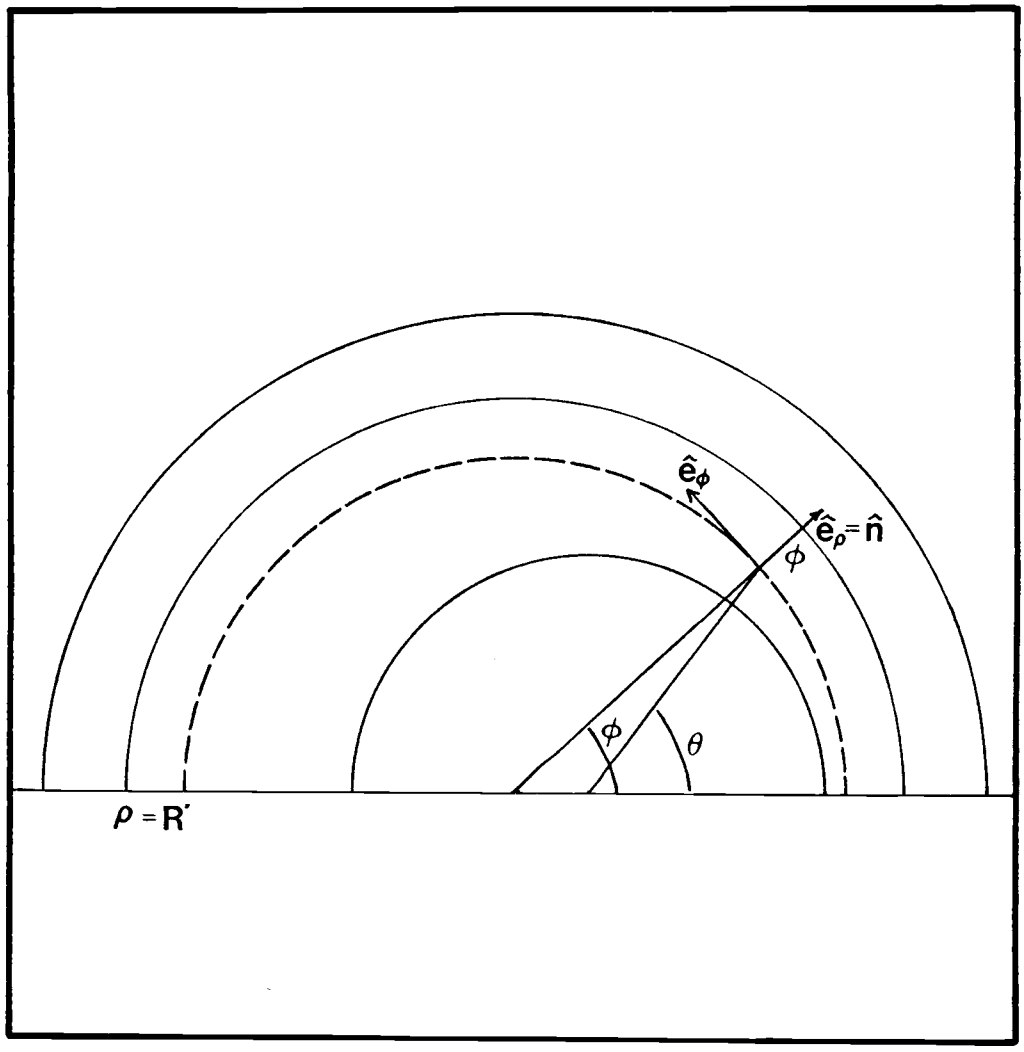


Figure 4. Unit vector normal to the $\rho = R'$ boundary.



heat flux vectors from the two coordinate systems. The heat flux vectors are given as

$$\bar{q}(r,\theta) = -k\ \bar{\nabla}\ T(r,\theta)$$

$$= -k(\hat{e}_r\ \partial/\partial r\ T(r,\theta) + \hat{e}_\theta\ 1/r\ \partial/\partial\theta\ T(r,\theta)) \quad 2.6$$

and

$$\bar{q}(\rho,\phi) = -k\ \bar{\nabla}\ T(\rho,\phi)$$

$$= -k(\hat{e}_p\ \partial/\partial\rho\ T(\rho,\phi) + \hat{e}_\phi\ 1/\rho\ \partial/\partial\phi\ T(\rho,\phi)) \quad 2.7$$

where k = conductivity of the fuel

$\hat{e}_\rho, \hat{e}_r, \hat{e}_\theta, \hat{e}_\phi$ = unit vectors in their respective directions

$T(r,\theta), T(\rho,\phi)$ = temperature for their respective coordinate systems

To facilitate the dot product calculation, the unit vectors describing the two coordinate systems are translated into a common (x,y) coordinate system.

$$\hat{e}_\phi = \hat{e}_y \cos\phi - \hat{e}_x \sin\phi \quad 2.8$$

$$\hat{e}_\rho = \hat{e}_x \cos\phi + \hat{e}_y \sin\phi \quad 2.9$$

$$\hat{e}_r = \hat{e}_x \cos\theta + \hat{e}_y \sin\theta \quad 2.10$$

$$\hat{e}_\theta = \hat{e}_y \cos\theta - \hat{e}_x \sin\theta \quad 2.11$$

The dot products are then calculated as

$$\hat{e}_\rho \cdot \bar{q}(r,\theta)\Big|_{r = \frac{R' \sin \phi}{\sin \theta}} = \hat{e}_\rho \cdot \bar{q}(\rho,\phi)\Big|_{\rho = R'} \qquad 2.12$$

Using the relationships described above, the left hand side becomes:

$$(\hat{e}_x \cos \phi + \hat{e}_y \sin \phi)$$

$$\cdot -k[(\hat{e}_x \cos \theta + \hat{e}_y \sin \theta)\ \partial/\partial r\ T(r,\theta) \qquad 2.13$$

$$+ 1/r\ (\hat{e}_y \cos \theta - \hat{e}_x \sin \theta)\ \partial/\partial \theta\ T(r,\theta)]$$

which when evaluated yields

$$-k[\cos(\phi-\theta)\,\partial/\partial r\ T(r,\theta) + 1/r \sin(\phi-\theta)\ \partial/\partial \theta\ T(r,\theta)] \qquad 2.14$$

The right hand side is easily determined to be:

$$-k\ \partial/\partial \rho\ T(\rho,\phi) \qquad 2.15$$

As the angles pass from zero to π along the boundary R', the maximum difference between the two angles $(\phi-\theta)$, near $\pi/2$, is found to be small. Using the following approximations

$$\sin(x) \simeq 0 \qquad 2.16$$

$$\cos(x) \simeq 1 \qquad 2.17$$

where x = small angle

allows equation 2.14 to be simplified to the following:

$$-k[\cos(\phi-\theta)\ \partial/\partial r\ T(r,\theta) + 1/r\ \sin(\phi-\theta)\ \partial/\partial\theta\ T(r,\theta)] \tag{2.18}$$

$$\simeq -k\ \partial/\partial r\ T(r,\theta)$$

The introduction of a term describing the gap width as a function of angle allows the following finite difference approximation of the eccentric conditions:

$$\bar{q} \simeq -k\ \frac{\Delta T}{G(\phi)} \tag{2.19}$$

where T = temperature difference across gas gap

$G(\phi)$ = gap width as a function of angle

A standard finite difference code, restricted to concentric regions, utilizes the following approximation for the eccentric conditions:

$$\bar{q} \simeq -k(\phi)\ \frac{\Delta T}{\Delta R} \tag{2.20}$$

where $k(\phi)$ = angular dependent conductivity

ΔT = temperature difference across gas gap

ΔR = width of gas gap (constant)

Equating these two approximations gives the equation which describes the angular dependent conductivity within the gas gap.

$$k(\phi) = k\ \frac{\Delta R}{G(\phi)} \tag{2.21}$$

The term, $G(\phi)$, which describes the width of the gas gap as a function of angle is calculated by taking the following difference

$$G(\phi) = \rho\Big|_{R_c} - \rho\Big|_{R_f} \qquad 2.22$$

where $\rho\Big|_{R_c}$ = distance to the outer edge of gas gap

$\rho\Big|_{R_f}$ = distance to pellet surface (see equation 2.4)

Combining these equations results in the gas gap as a function of angle being given by the following:

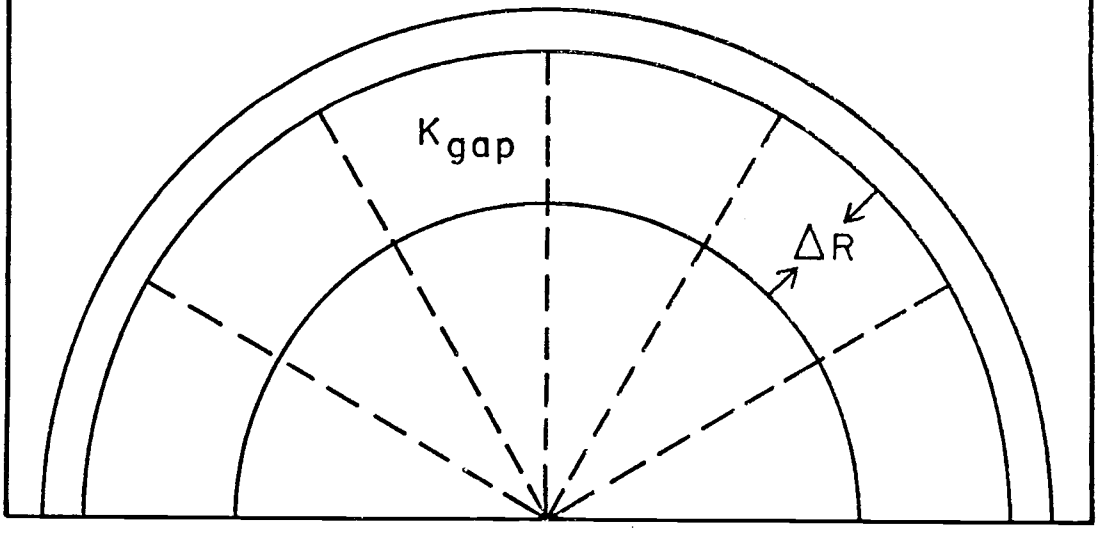
$$G(\phi) = R_c - [d \cos \phi + (R_f^2 - d^2[1 - \cos^2 \phi])^{\frac{1}{2}}] \qquad 2.23$$

The modeled conductivity program, utilizing the relationships derived in the above equations, results in the finite difference geometry as shown in Figure 5. The fuel pellet is centered within the cladding and the variable conductivity within the gas gap now accounts for the eccentricity of the problem. The modeled conductivity geometry places no restrictions on the required amount of radial or angular node regions to properly define the three regions of the problem. In addition, the modeled conductivity approach requires no translation of the generated fuel pellet temperatures to a separate coordinate system.

Figure 5. Modeled conductivity approximation.

$$K(\phi) = K_{gap} \frac{\Delta R}{G(\phi)}$$

$$G(\phi) = R_c - \left[d\cos\phi + (R_f^2 - d^2(1-\cos^2\phi))^{1/2} \right]$$



A finite difference code, MDL, was written to perform the operations required by the modeled conductivity approach. This code is described in Appendix B.

III. COMPARISONS, RESULTS, AND CONCLUSIONS

Comparison Tests

The comparisons between the ratchet boundary model and the modeled conductivity model utilized a set of six eccentricities ranging from 5%-99%. Three gap width to fuel diameter ratios, varying from 1.22% to 15.63% were used. Each problem contained three separate regions; fuel (conductivity = 0.025 W/cm°C), cladding (conductivity = 0.15 W/cm°C), and gas gap (conductivity = 0.004 W/cm°C). Both models were divided into 15 azimuthal node regions. The ratchet boundary model required 22 radial node regions as compared to only seven radial node regions for the modeled conductivity program.

The main points of interest in this comparison study were the following:

1) Comparison of cladding temperatures as predicted by the two models.
2) Comparison of fuel pellet temperatures as predicted by the two models.
3) Comparison of economics between the two models as predicted by such factors as iterations and CPU time required for convergence.

Results

2.1 Cladding Temperatures

The difference in maximum cladding temperatures as predicted by the two models, for all cases, are shown in Figures 6-8. The percent deviation is derived from the following equation:

$$\% \text{ deviation} = \frac{T_{MDL_i} - T_{RAT_i}}{T_{MDL_i}} \times 100 \qquad 3.1$$

where T_{MDL_i} = the temperature predicted by the modeled conductivity program for angle i

T_{RAT_i} = the temperature predicted by the ratchet boundary program for angle i

The results show that the cases utilizing a fuel pin geometry similar to current light water reactor pin geometries (Figure 6) show less than 0.1% deviation in the maximum cladding temperatures as predicted by the two models. In most cases the maximum deviation in cladding temperature predictions, between the two models, is less than 1.0% and is usually less than 0.67%.

The nodal points that lie on the small gas gap side of the problem, for all cases of very high eccentricity, show a deviation from the standard pattern, as shown by the dotted lines in Figures 6-8. This deviation is thought to be caused from the geometrical factors inherent

Figure 6. Percent deviation in maximum cladding temperatures (gap-width-to-fuel-diameter = 0.0122).

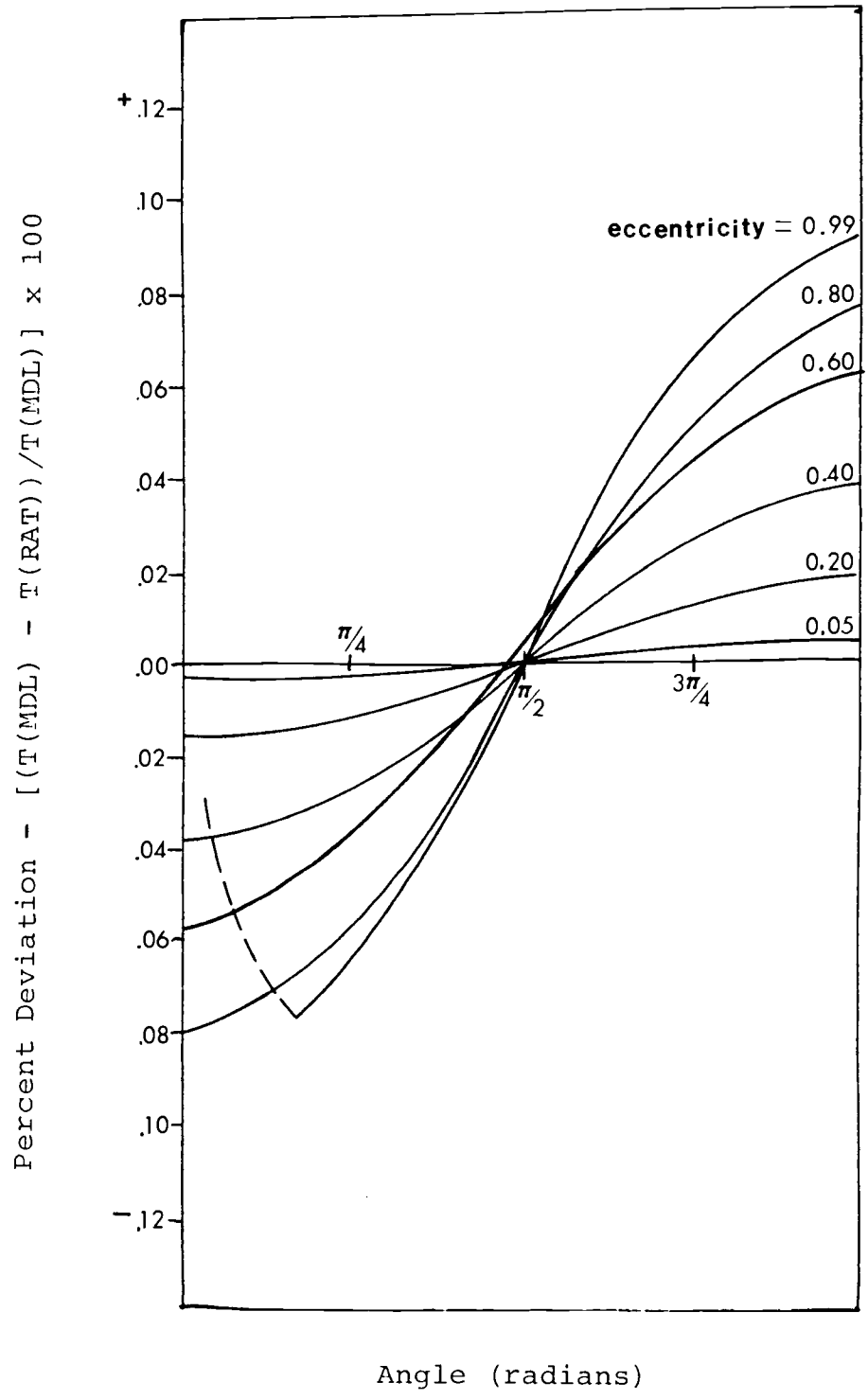


Figure 7. Percent deviation in maximum cladding temperatures (gap-width-to-fuel-diameter = 0.0676).

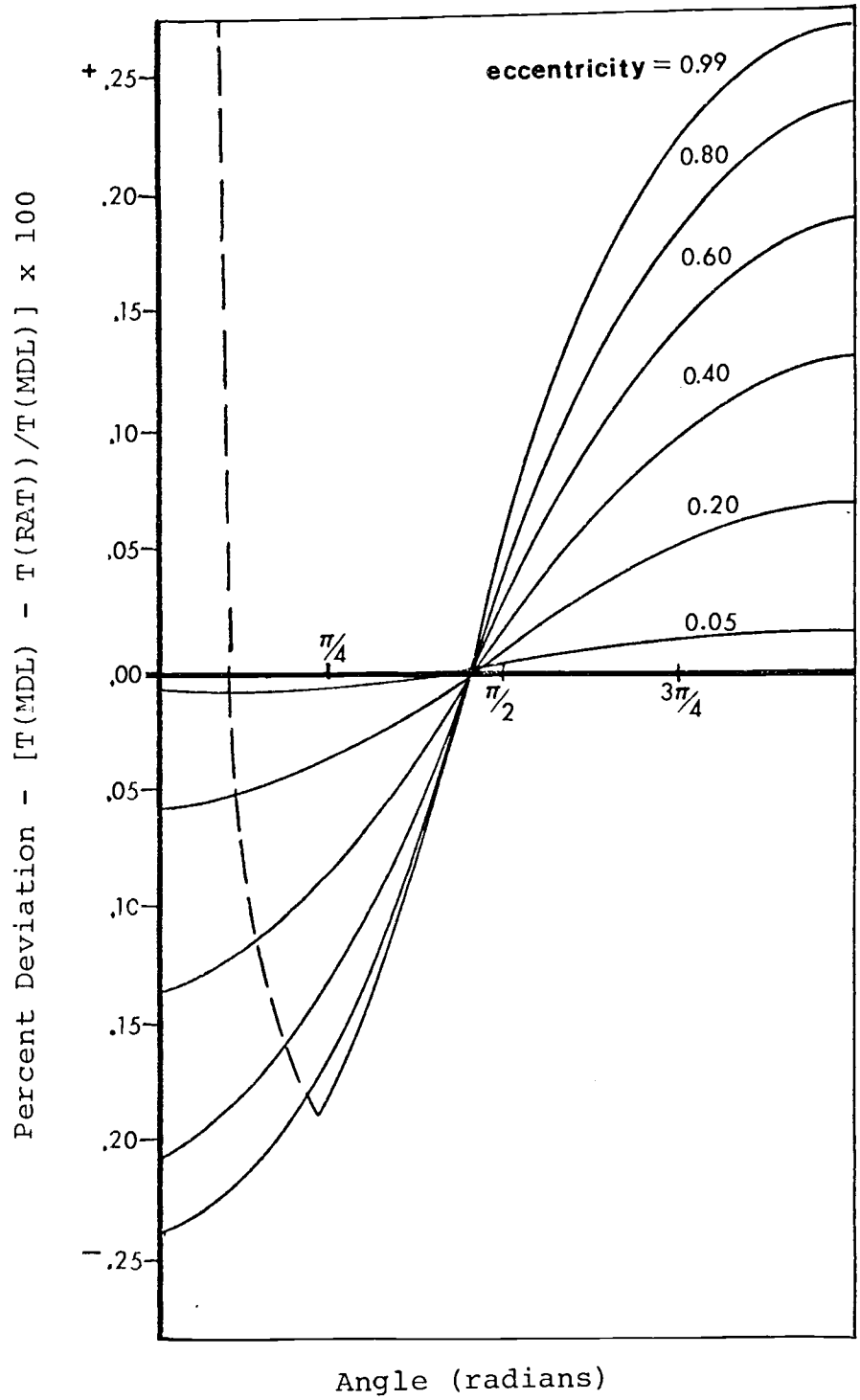
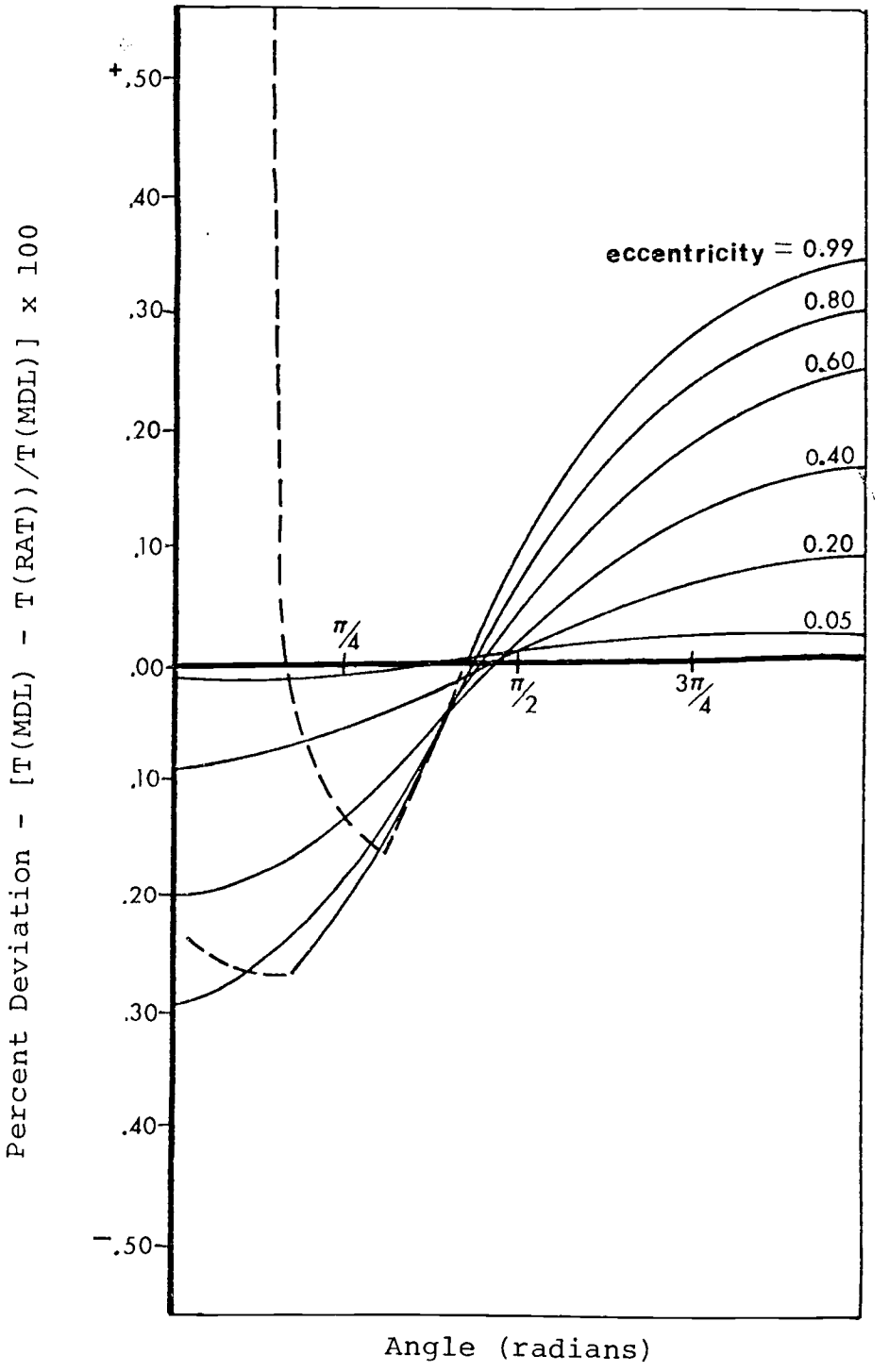


Figure 8. Percent deviation in maximum cladding temperatures (gap-width-to-fuel-diameter = 0.1563).



in the ratchet boundary method. The ratchet boundary method, in essence, presents an averaged distance between the fuel pellet surface and the cladding. This averaged distance does not take into account the dominance that the points lying nearest to the cladding play in the problem. In the cases of high eccentricity, were large amounts of heat are transferred azimuthally through the fuel pellet, the points lying next to the cladding present a path of very small resistance to heat transfer. As the gas gap becomes larger and the eccentricity higher, these points tend to completely dominate as the heat transfer path from the fuel pellet surface to the cladding. This dominance, as predicted by the modeled conductivity model, is reflected in the larger deviations of cladding temperatures at these points.

2.2 Fuel Pellet Temperatures

The temperature profiles generated for the fuel pellet by the two models are shown in Figures 9-17. The isotherms for the cases utilizing a fuel pin geometry similar to current light water reactor pin geometries (Figures 9-11) shows excellent correspondence between the two models. The deviation of temperature profiles, as shown by the plotted isotherms, indicate good agreement at low eccentricities and varies as a function of eccentricity, with large eccentricities showing the maximum

FIGURE 9 FUEL PELLET ISOTHERMS

GWTFD RATIO=0.0122 ECCENTRICITY=0.05

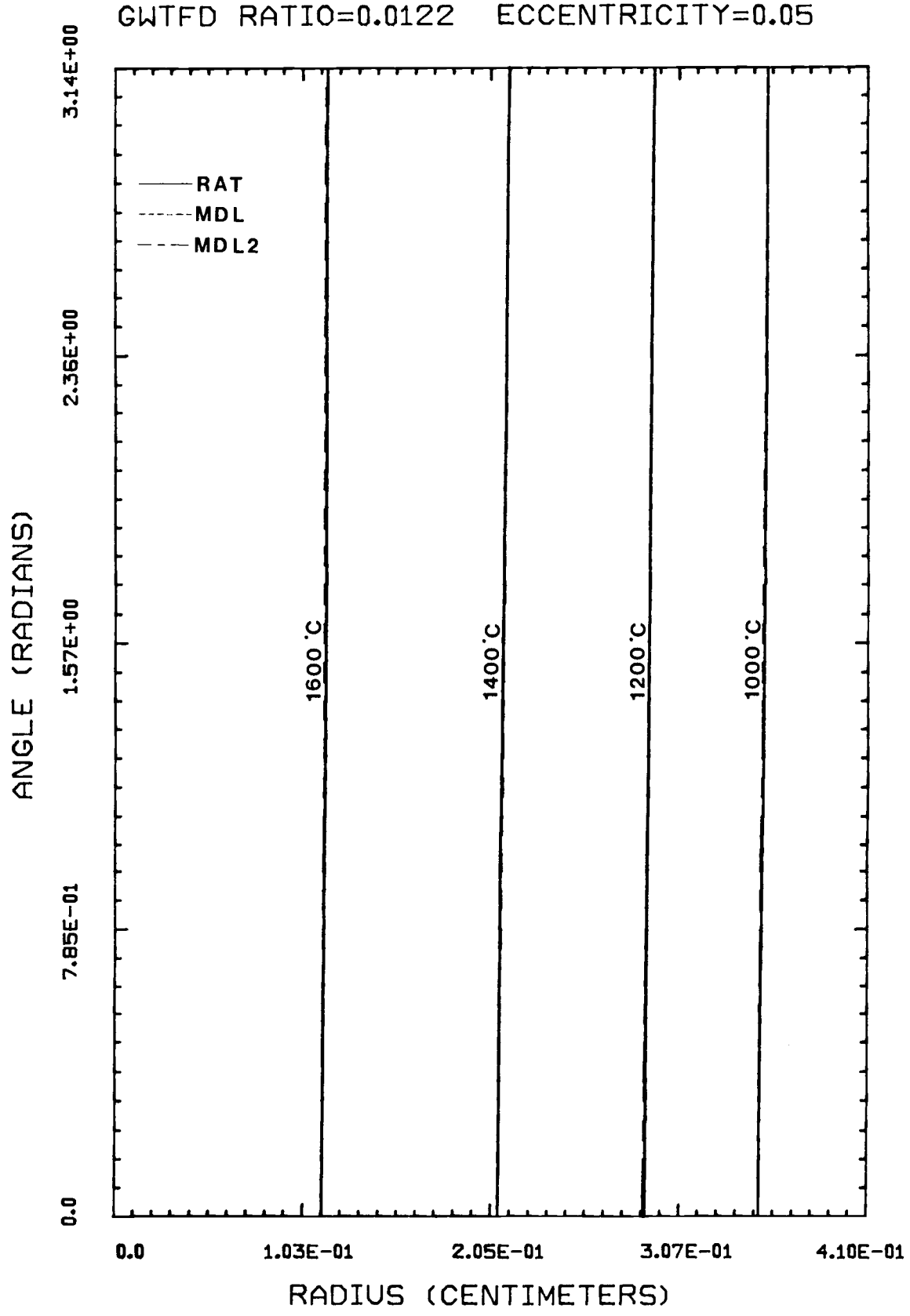


FIGURE 10 FUEL PELLET ISOTHERMS

GWTFD RATIO=0.0122 ECCENTRICITY=0.60

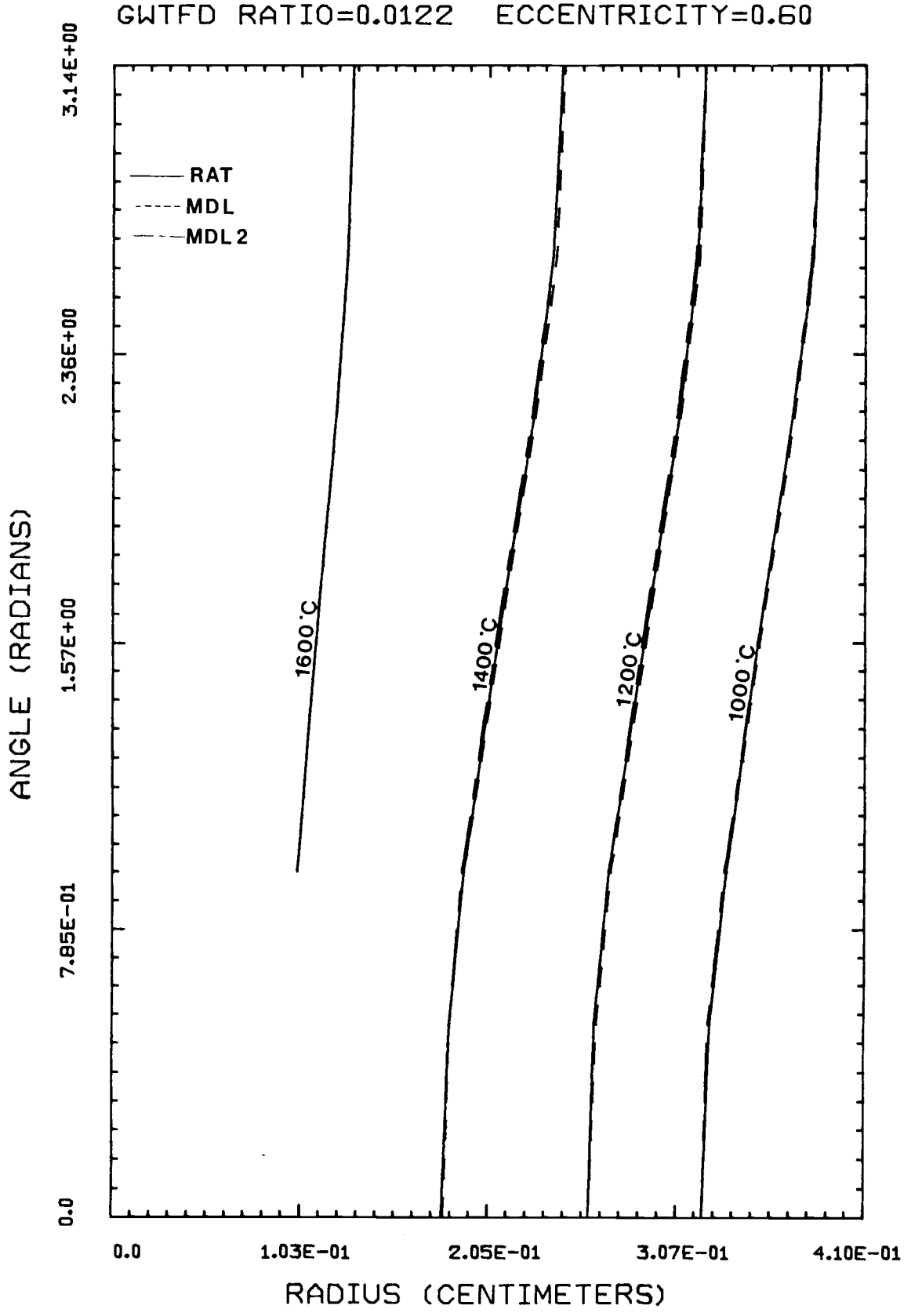


FIGURE 11 FUEL PELLET ISOTHERMS

GWTFD RATIO=0.0122 ECCENTRICITY=0.99

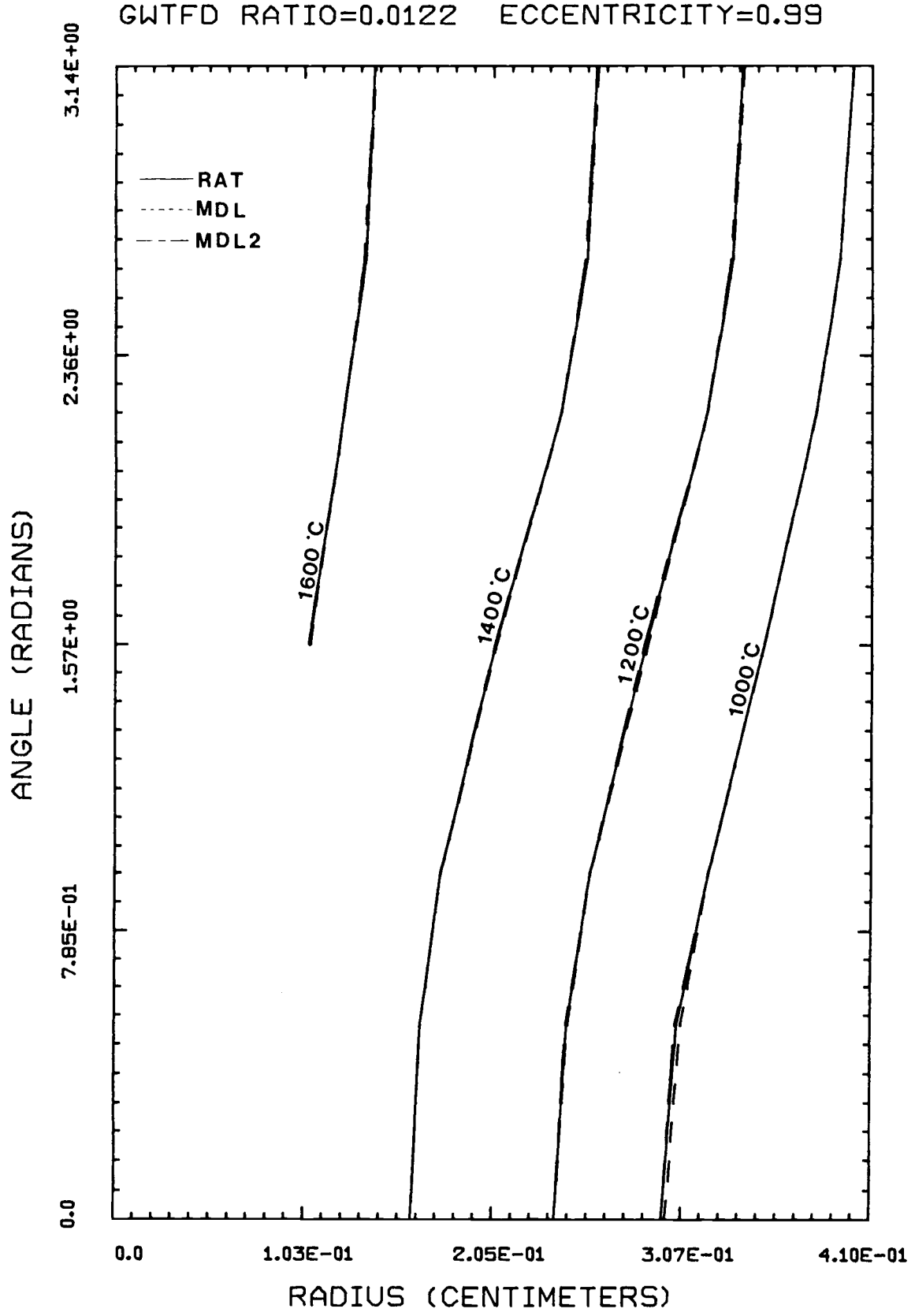


FIGURE 12 FUEL PELLET ISOTHERMS

GWTFD RATIO=0.0676 ECCENTRICITY=0.05

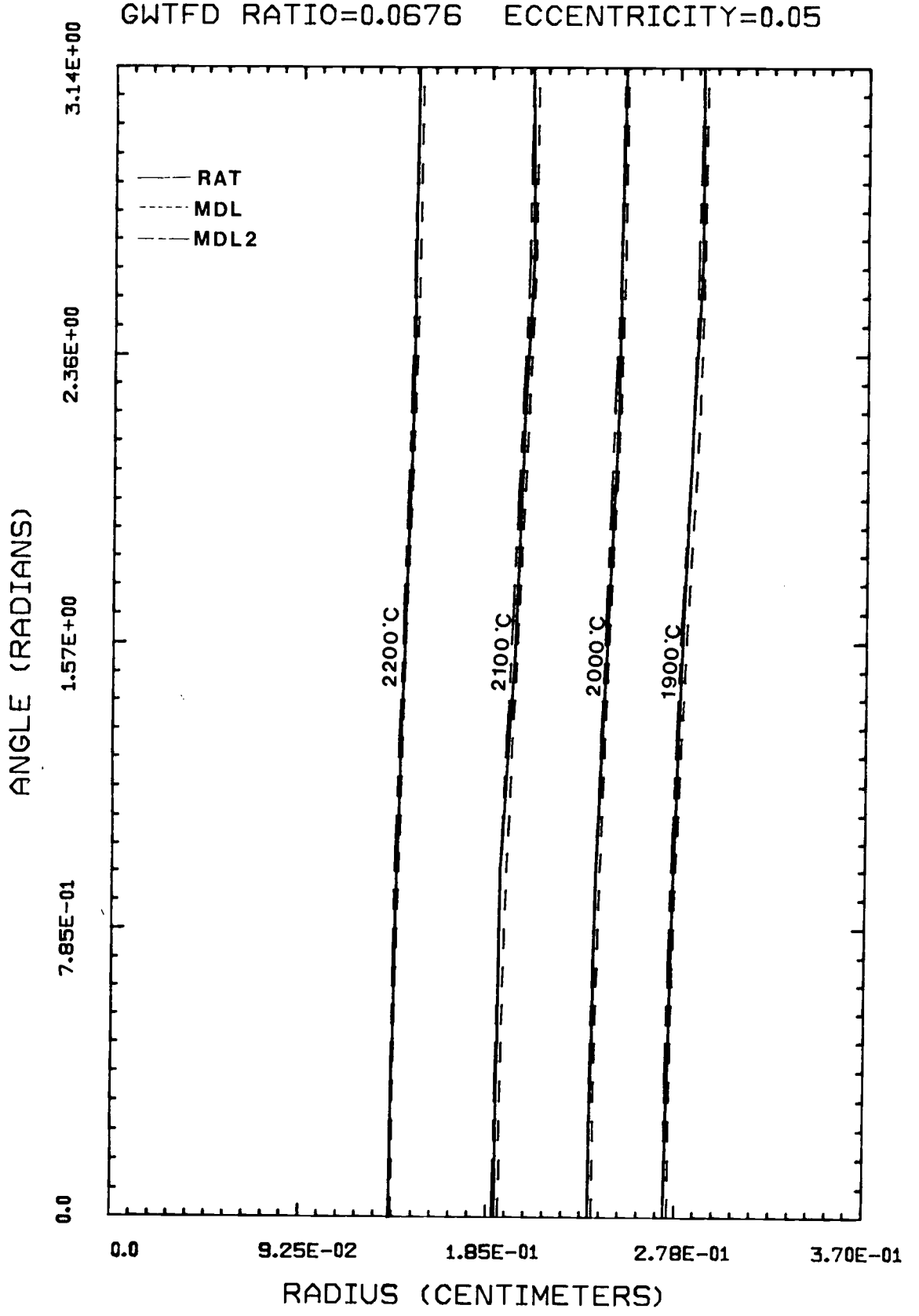


FIGURE 13 FUEL PELLET ISOTHERMS

GWTFD RATIO=0.0676 ECCENTRICITY=0.60

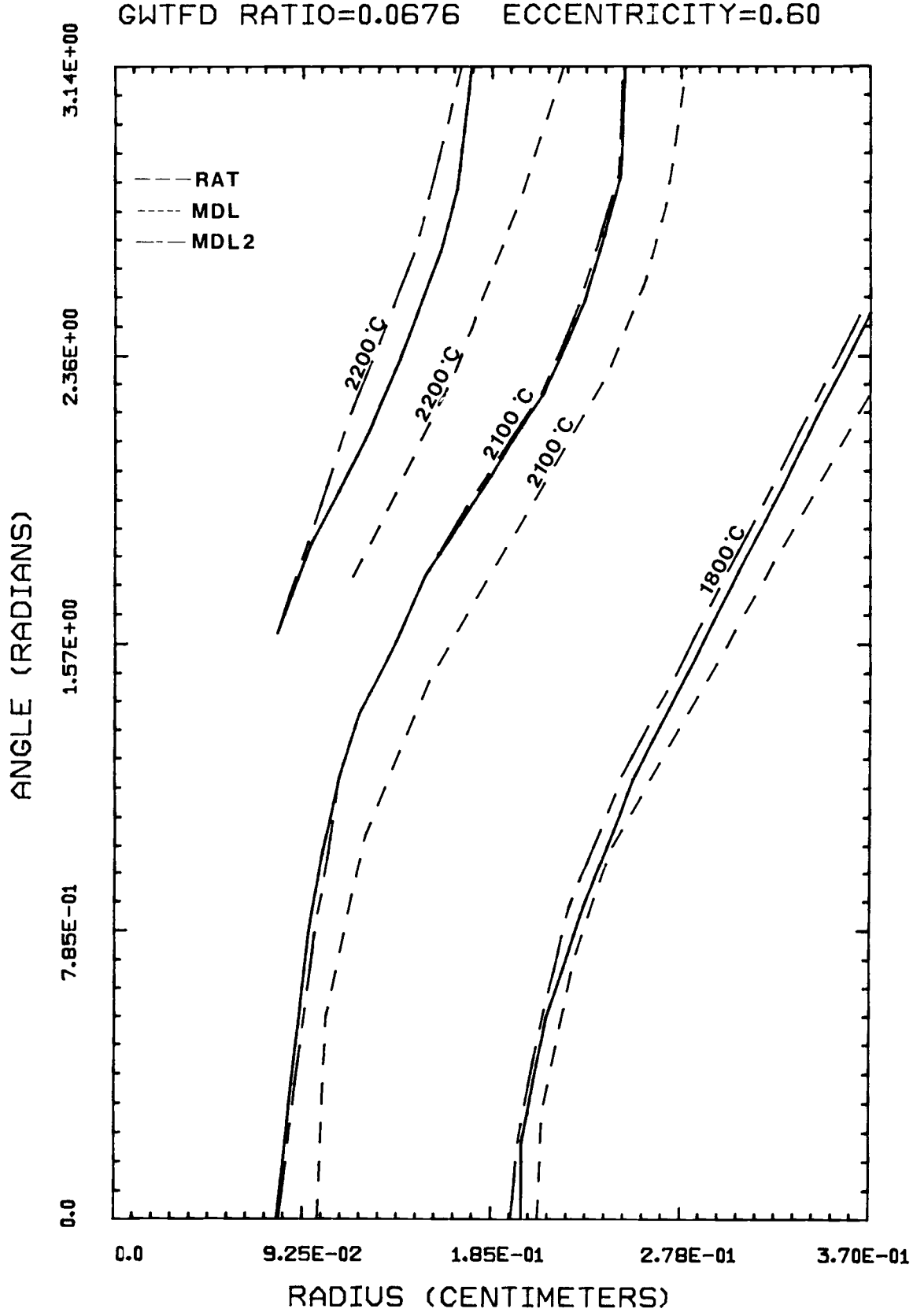


FIGURE 14 FUEL PELLET ISOTHERMS

GWTFD RATIO=0.0676 ECCENTRICITY=0.99

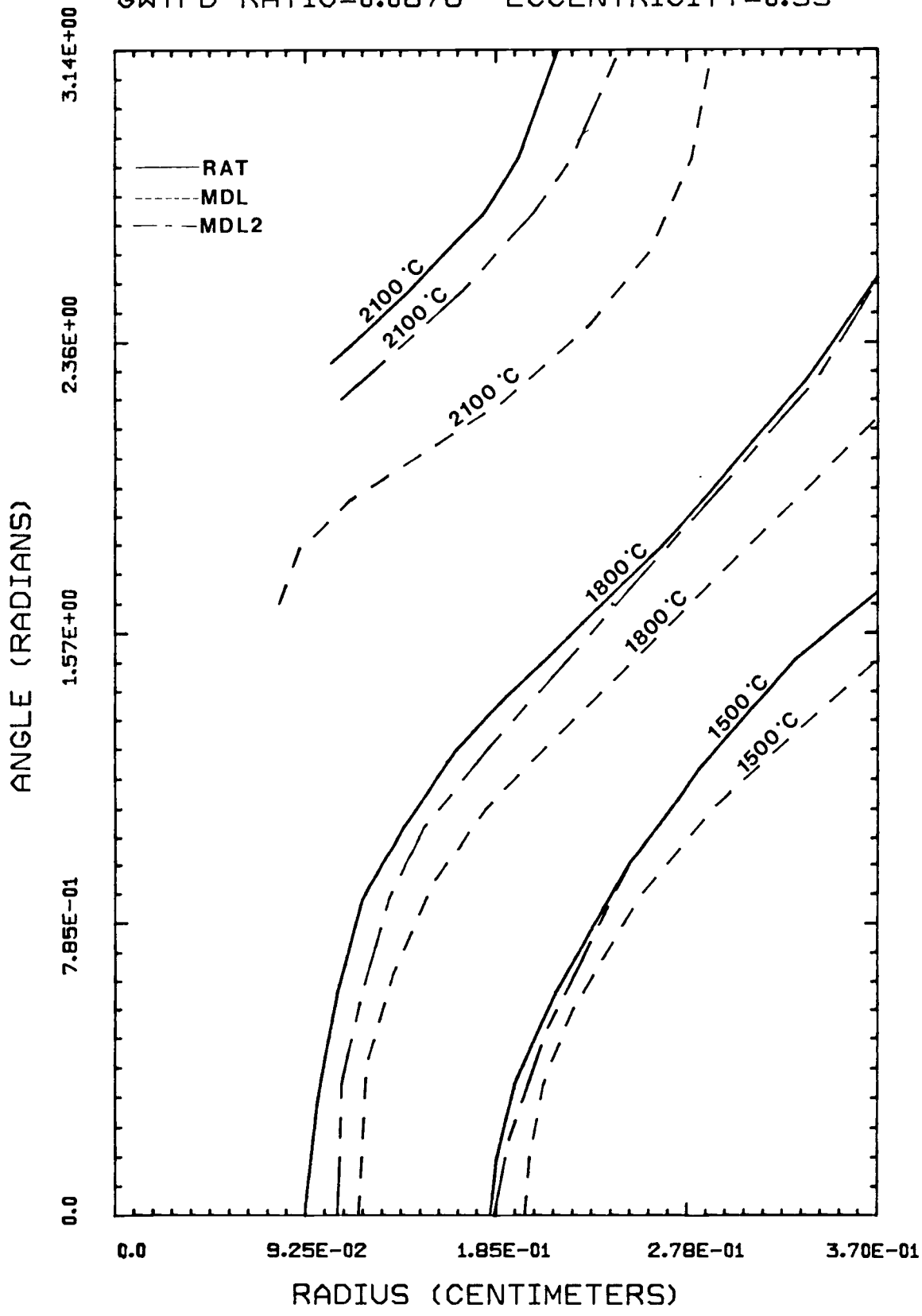


FIGURE 15 FUEL PELLET ISOTHERMS
GWTFD RATIO=0.1563 ECCENTRICITY=0.05

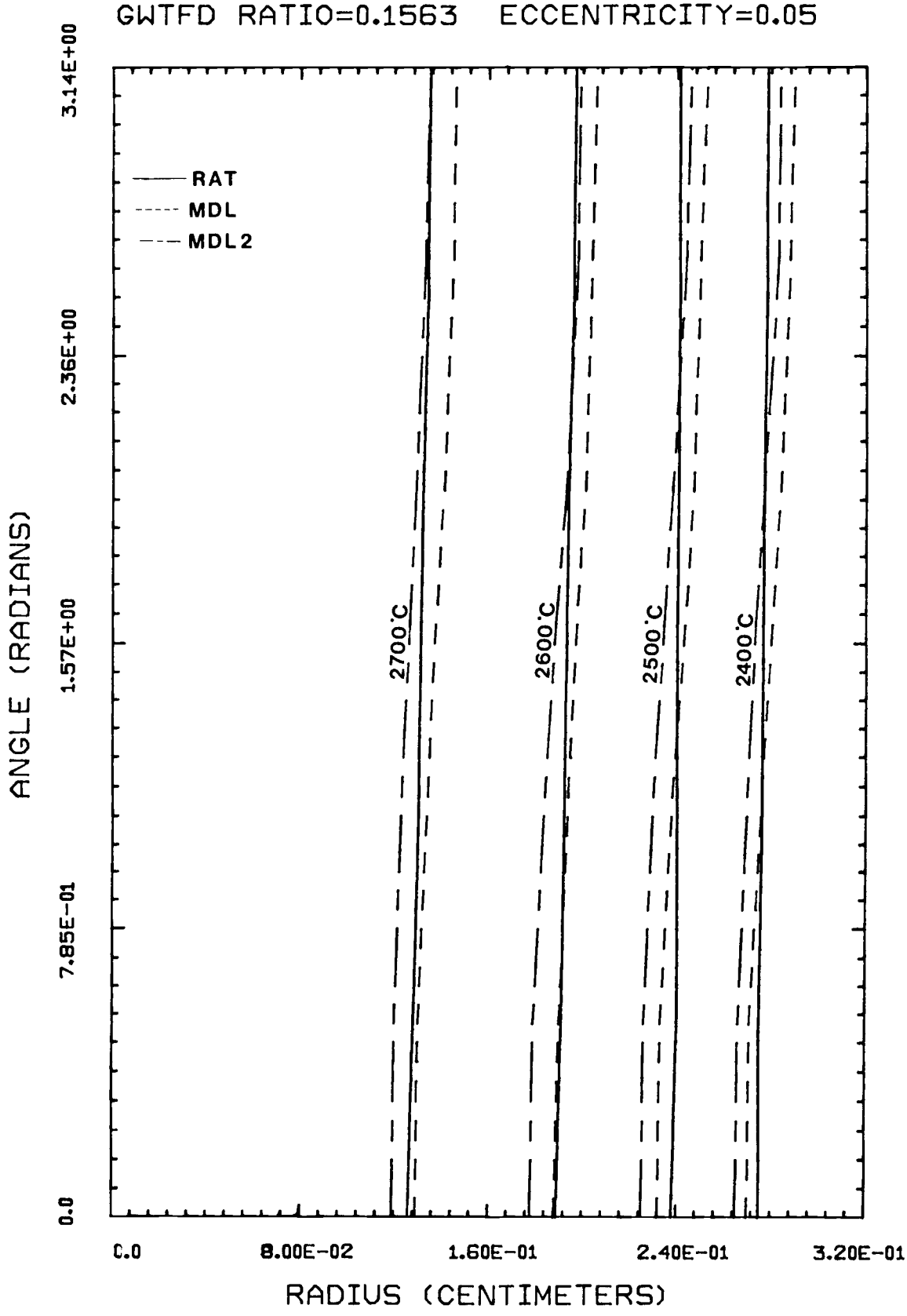


FIGURE 16 FUEL PELLET ISOTHERMS

GWTFD RATIO=0.1563 ECCENTRICITY=0.60

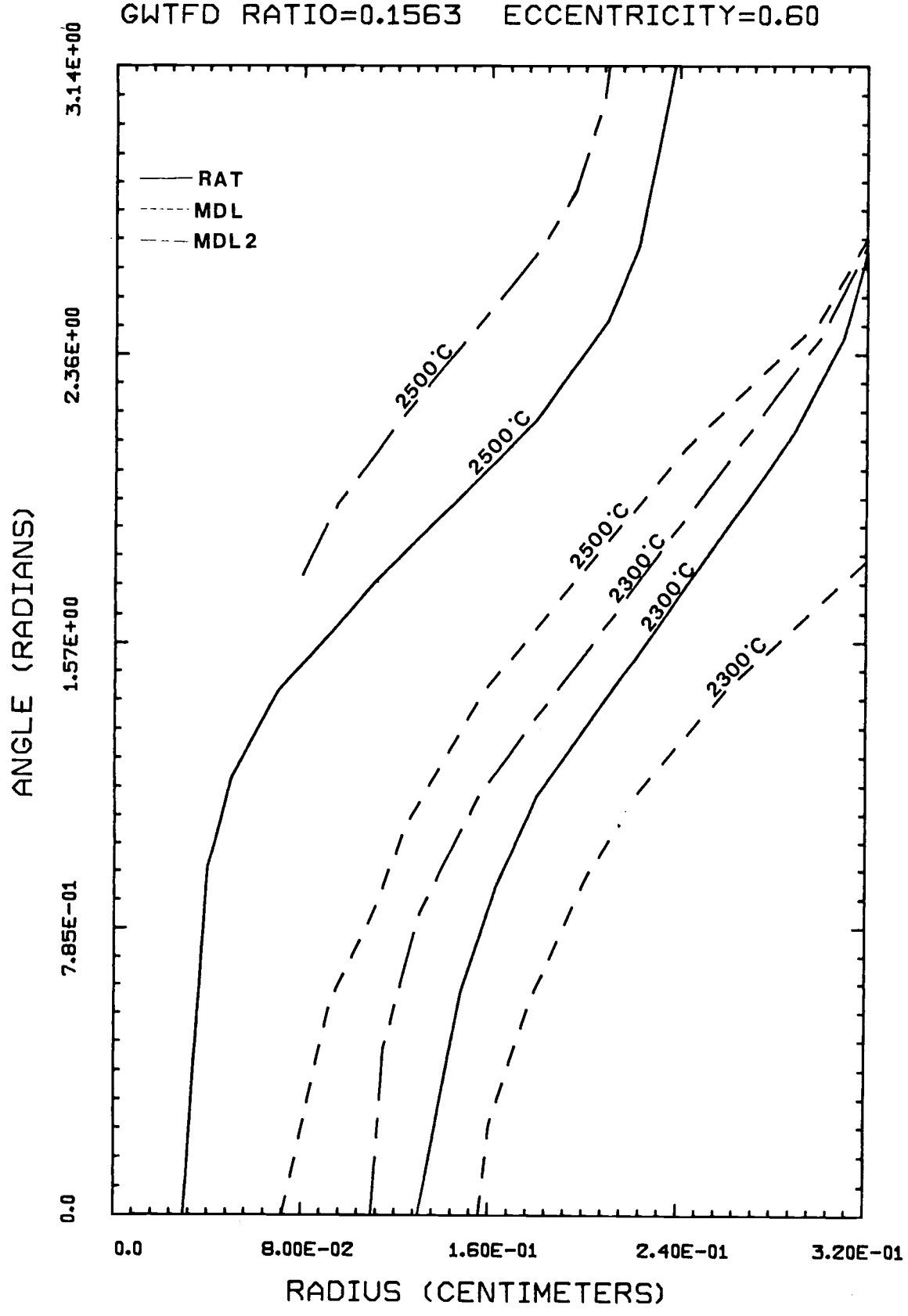
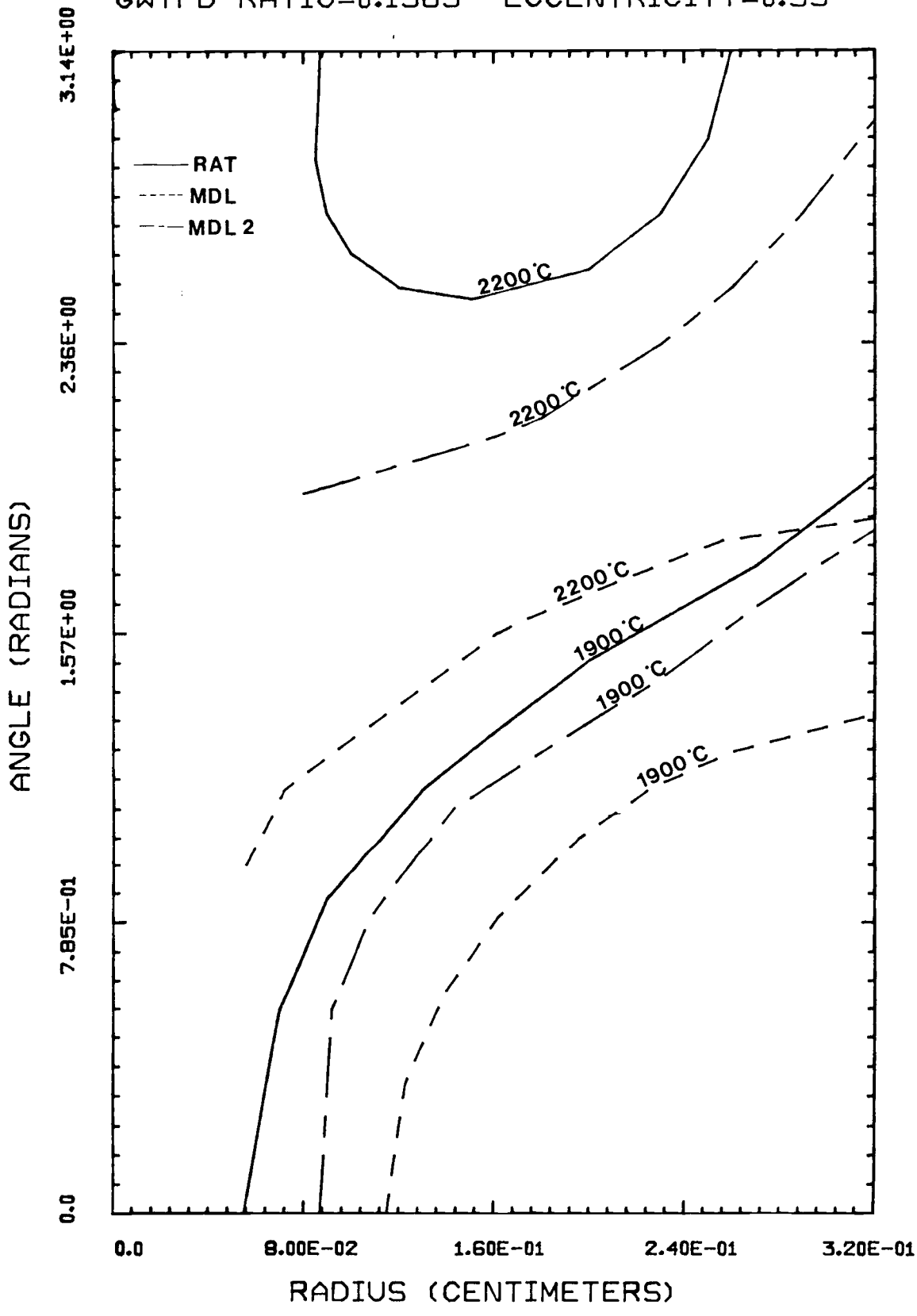


FIGURE 1 7 FUEL PELLET ISOTHERMS

GWTFD RATIO=0.1563 ECCENTRICITY=0.99



deviations in all cases. The temperature profiles are also seen to vary as a function of gap-width-to-fuel-diameter ratio with the cases utilizing a small ratio showing a greater correspondence between temperature profiles than the cases utilizing a high ratio for the same eccentricity.

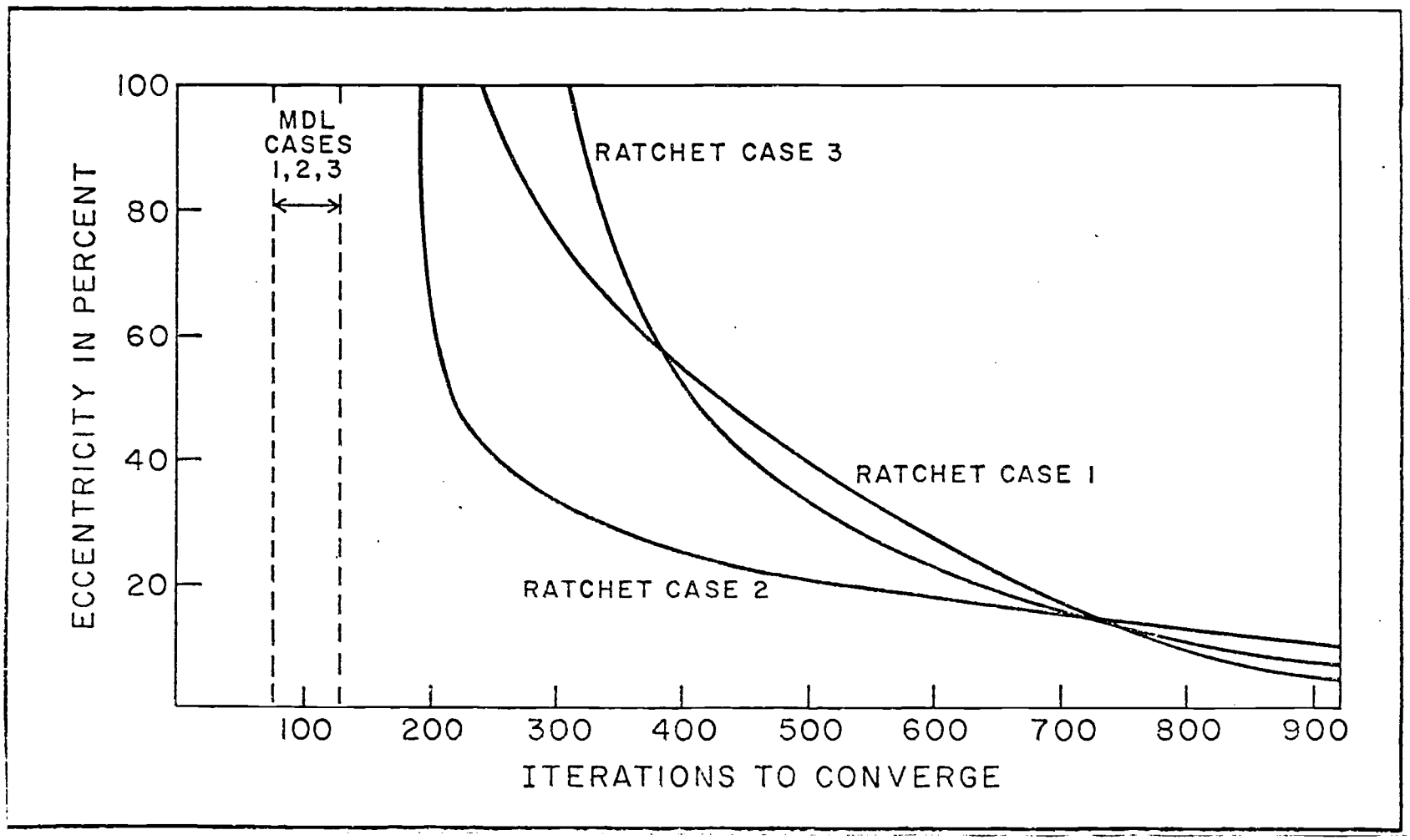
2.3 Economics

The amount of iterations required to reach a satisfactory convergence for the two models is shown in Figure 18.

The results show that in all cases the ratchet boundary model required more iterations per run than the modeled conductivity program. The ratio of iterations required for convergence shows a minimum of approximately 2 to 1 at high eccentricities to a maximum of 13 to 1 at low eccentricities for the two models. At low eccentricities the ratchet boundary program was shown to require a stricter convergence criteria than the modeled conductivity program to avoid anomalous convergence.

The amount of CPU time required for the two methods shows that it costs from 6 to 40 times as much to run the ratchet boundary program as it does to run the modeled conductivity program.

Figure 18. Required iterations to reach convergence.



Comparison Conclusions

The results of the comparison test indicate that both models are employing an effective means of transporting heat generated by a fuel pellet through the gas gap and into the cladding.

The modeled conductivity approach shows higher fuel temperatures than the ratchet boundary method. This is due, in large part, to the modeled conductivity approach neglecting the angular component of the heat flux vector from the fuel pellet surface. The negligence of this component of the heat flux vector tends to impose a conservative resistance to heat flow within the gas gap. As the eccentricity and gap-width-to-fuel-diameter ratio increases, the angular heat flux vector becomes larger which results in conservatively higher gas gap resistances. The higher resistances result in higher fuel pellet temperatures for the modeled conductivity approach.

In the cases where the angular heat flux vector may be considered negligible, i.e. very low eccentricities or small gap-width-to-fuel-diameter ratios (typical of light water reactors), or the interest is mainly in cladding temperatures, the modeled conductivity approach is shown to have a marked advantage over the ratchet boundary method due to the extra costs associated with the ratchet boundary method.

In the actual programming of these two methods, the ratchet boundary method creates some difficulty due to the geometry of the ratcheted fuel pellet surface. The very small radial mesh spacing created by the ratcheted fuel pellet boundary along with the overall number of node points required, leads to the necessity for more iterations and thus higher costs to reach convergence.

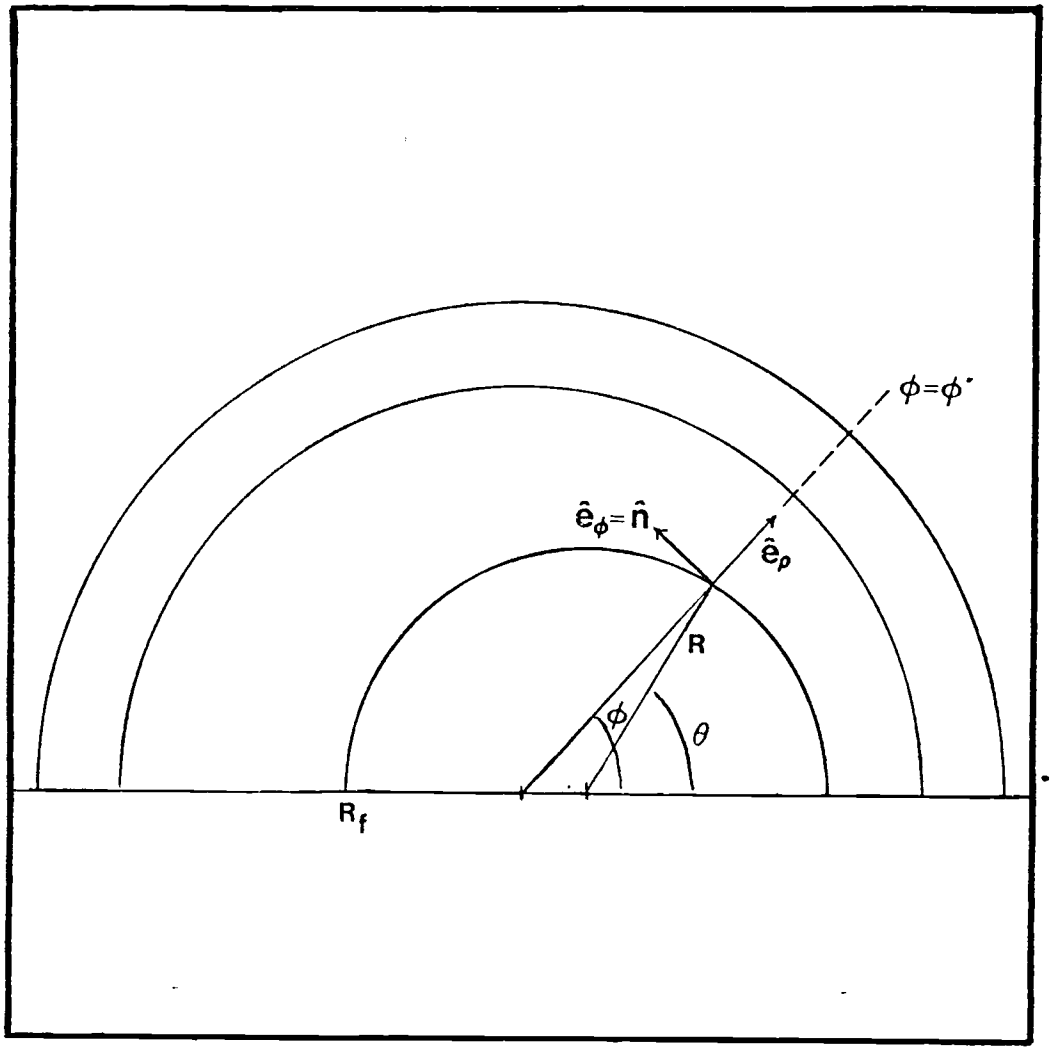
IV. IMPROVED MODELED CONDUCTIVITY

The previous modeled conductivity approach was shown to generate temperatures within the fuel pellet which are higher, in most cases, than those generated by the ratchet boundary model. This temperature difference was thought to be due, in large part, to the negligence of the angular heat flux vector from the fuel pellet surface. The improved modeled conductivity takes this additional source of conductance into account and generates a total heat flux vector from the fuel pellet surface which includes both the angular and radial components.

The radial component was found by dotting the normal vector at the $\rho = R'$ boundary into the total heat flux vector for the fuel pellet (see equation 2.12 and Figure 4). In a like manner, the angular component is found by dotting the normal vector at $\phi = \phi'$ boundary into the total heat flux vector for the fuel pellet (see Figure 19). The normal vector to the $\phi = \phi'$ boundary is given by $\hat{n}$ which is equal to the unit vector in the ϕ-direction, $\hat{e}_\phi$. The heat flux fectors for the two coordinate systems are given by equations 2.6 and 2.7 which are listed below:

$$\begin{aligned} \bar{q}(r,\theta) &= -k\ \bar{\nabla}\ T(r,\theta) \\ &= -k(\hat{e}_r\ \partial/\partial r\ T(r,\theta) + \hat{e}_\theta\ 1/r\ \partial/\partial\theta\ T(r,\theta)) \end{aligned} \qquad 2.6$$

Figure 19. Unit vector normal to the $\phi = \phi'$ boundary.



and

$$\bar{q}(\rho,\phi) = -k\ \bar{\nabla}\ T(\rho,\phi)$$

$$= -k(\hat{e}_\rho\ \partial/\partial\rho\ T(\rho,\phi) + \hat{e}_\phi\ 1/\rho\ \partial/\partial\phi\ T(\rho,\phi)) \qquad 2.7$$

To facilitate the dot product calculation, the unit vectors describing the two coordinate systems are once again translated into a common (x,y) coordinate system as given by equations 2.8-11 which are listed below:

$$\hat{e}_\phi = \hat{e}_y \cos\phi - \hat{e}_x \sin\phi \qquad 2.8$$

$$\hat{e}_\rho = \hat{e}_x \cos\phi + \hat{e}_y \sin\phi \qquad 2.9$$

$$\hat{e}_r = \hat{e}_x \cos\theta + \hat{e}_y \sin\theta \qquad 2.10$$

$$\hat{e}_\theta = \hat{e}_y \cos\theta - \hat{e}_x \sin\theta \qquad 2.11$$

The dot products are now calculated as:

$$\hat{e}_\phi \cdot \hat{q}(r,\theta)\Big|_{\theta=\sin^{-1}(\frac{\rho \sin\phi'}{r})} = \hat{e}_\phi \cdot q(\rho,\phi)\Big|_{\phi=\phi'} \qquad 4.1$$

Using the relationships in equations 2.8-11, the left hand side becomes:

$$(\hat{e}_y \cos\phi - \hat{e}_x \sin\phi)$$

$$\cdot\ -k(\hat{e}_x \cos\theta + \hat{e}_y \sin\theta)\ \partial/\partial r\ T(r,\theta) \qquad 4.2$$

$$+\ 1/r(\hat{e}_y \cos\theta - \hat{e}_x \sin\theta)\ \partial/\partial\theta\ T(r,\theta))$$

which when evaluated yields:

$$-k(\sin(\theta-\phi)\ \partial/\partial r\ T(r,\theta) + 1/r\ \cos(\theta-\phi)\ \partial/\partial\theta\ T(r,\theta)) \quad 4.3$$

The right hand side is easily determined to be:

$$-k(1/\rho\ \partial/\partial\phi\ T(p,\phi)) \quad 4.4$$

The total heat flux vector may now be found by an addition of the angular and radial components. The components of the heat flux vector normal to the $\rho = \rho'$ boundary are given by equations 2.14 and 2.15.

$$-k[\cos(\phi-\theta)\ \partial/\partial r\ T(r,\theta) + 1/r\ \sin(\phi-\theta)\ \partial/\partial\theta\ T(r,\theta)]$$

2.14 and 2.15

$$= -k\ \partial/\partial\rho\ T(\rho,\phi)$$

Combining radial and angular components and utilizing the following relationships:

$$\cos(A-b) = \cos(b-A) \quad 4.5$$

$$\sin(A-b) = -\sin(b-A) \quad 4.6$$

The total heat flux vector is found to be:

$$-k[[\cos(\theta-\phi) + \sin(\theta-\phi)]^2(\partial/\partial r\ T(r,\theta))^2 + (1/r)^2[\cos(\theta-\phi) - \sin(\theta-\phi)]^2(\partial/\partial\theta\ T(r,\theta))^2]^{\frac{1}{2}} \quad 4.7$$

The introduction of a term describing the gap width as a function of angle, as given by equation 2.23, allows the

following finite difference approximation:

$$\bar{q} = -k[[\cos(\theta-\phi) + \sin(\theta-\phi)]^2 \left(\frac{\Delta T_R}{G(\phi)}\right)^2 + (1/r)^2[\cos(\theta-\phi) - \sin(\theta-\phi)]^2 \left(\frac{\Delta T_\theta}{\Delta\theta}\right)^2]^{\frac{1}{2}} \quad 4.8$$

where $G(\phi)$ = gap width as a function of angle

$\Delta\theta$ = angular distance between node points

ΔT_R = radial temperature difference

ΔT_θ = angular temperature difference

Equating this approximation to the standard finite difference equation, as given by 2.20, leads to the equation which describes the angular dependent conductivity within the gas gap.

$$k(\phi) = k[[\cos(\theta-\phi) + \sin(\theta-\phi)]^2 \left[\frac{\Delta R}{G(\phi)}\right]^2 + (1/r)^2[\cos(\theta-\phi) - \sin(\theta-\phi)]^2 \left(\frac{\Delta R}{\Delta\theta}\right)^2\left(\frac{\Delta T_\theta}{\Delta T_R}\right)^2]^{\frac{1}{2}} \quad 4.9$$

Utilizing the fact that the radial temperature difference, between node points, across the gas gap width, ΔT_R, is very much larger than the angular temperature difference, between node points on the fuel pellet surface, ΔT_θ, leads to the following approximations:

$$\Delta T_R >> \Delta T_\theta \quad 4.10$$

$$[\cos(\theta-\phi) + \sin(\theta-\phi)]^2 \left(\frac{\Delta R}{G(\phi)}\right)^2$$

$$>> (1/r)^2 [\cos(\theta-\phi) - \sin(\theta-\phi)]^2 \left(\frac{\Delta R}{\Delta\theta}\right)^2 \left(\frac{\Delta T_\theta}{\Delta T_R}\right)^2 \qquad 4.11$$

$$k(\phi) \simeq -k[\cos(\theta-\phi) + \sin(\theta-\phi)] \frac{\Delta R}{G(\phi)} \qquad 4.12$$

The improved modeled conductivity approach utilizes the same geometry as the modeled conductivity method. The difference in the two methods occurs in the angular dependent conductivity within the gas gap. The conductivity is slightly larger in the improved method due to the additional angular heat flux component from the fuel pellet surface. The bracketed term in equation 4.12 is now greater than unity in all nodal regions. As in the original modeled conductivity method, there are no restrictions on the required amount of radial or angular node points required.

A finite difference code, MDL - 2, was written to perform the operations required by the improved modeled conductivity model. This code is described in Appendix B.

V. COMPARISON AND RESULTS

Comparison Test

As in the previous comparison test, the study between the improved modeled conductivity model and the previous models utilized the same set of six eccentricities, ranging from 5% to 99%, for each of three gap-width-to-fuel-diameter ratios, varying from 1.22% to 15.62%. The fuel pin contained three separate regions; fuel (conductivity = 0.025 W/cm-°C), cladding (conductivity = 0.15 W/cm-°C), and gas gap (conductivity = 0.004 W/cm-°C). The fuel pin was once again divided into 15 azimuthal and 7 radial node regions.

The main points of interest were:

1) Comparison of cladding temperatures as predicted by the separate models.
2) Comparison of fuel pellet temperatures as predicted by the separate models.
3) Comparison of economics between the separate models as predicted by such factors as iterations and CPU time required for convergence.

Results

5.1 Cladding Temperatures

The differences in maximum cladding temperatures between the improved modeled conductivity approach and the ratchet boundary approach, for all cases, are shown in Figures 20-22. The percent deviation is once again given by:

$$\% \text{ Deviation} = \frac{T_{MDL-2_i} - T_{RAT_i}}{T_{MDL-2_i}} \times 100 \qquad 5.1$$

where T_{MDL-2_i} = the temperature predicted by the improved modeled conductivity program for angle i

T_{RAT_i} = the temperature predicted by the ratchet boundary program for angle i

The results show no significant change in the maximum cladding temperatures as predicted by the improved modeled conductivity approach over those generated by the original modeled conductivity approach. In the cases utilizing a fuel pin geometry similar to current light water reactor geometries (case 1, Figure 20), there is less than 0.1% deviation in the maximum cladding temperatures as predicted by the separate models. In most cases the maximum deviation in cladding temperature predictions is less than 1.0%.

The deviations in cladding temperature predictions for very high eccentricities shows, once again, the

dominance of those points lying very near the cladding as discussed in section 3.1 and predicted by both modeled conductivity methods.

5.2 Fuel Pellet Temperatures

The temperature profiles generated within the fuel pellet by the separate models are shown in Figures 9-17.

The plotted isotherms for the cases utilizing a fuel pin geometry most similar to current light water reactor pin geometries (case 1, Figures 9-11) shows excellent correspondence between all models. The deviation of temperature profiles, as shown by the plotted isotherms, indicate good agreement at low eccentricities and varies as a function of eccentricity, with large eccentricities showing the maximum deviations in all cases. The temperature profiles are seen to vary as a function of gap-width-to-fuel-diameter ratio. The cases utilizing a low gap-width-to-fuel-diameter ratio show a greater correspondence between temperature profiles than the cases utilizing a higher ratio for the same eccentricity.

The fuel pellet temperatures generated by the improved modeled conductivity approach are shown to correspond closer to the temperatures predicted by the ratchet boundary model than those of the original modeled conductivity model.

Figure 20. Percent deviation in maximum cladding temperatures (gap-width-to-fuel-diameter = 0.0122).

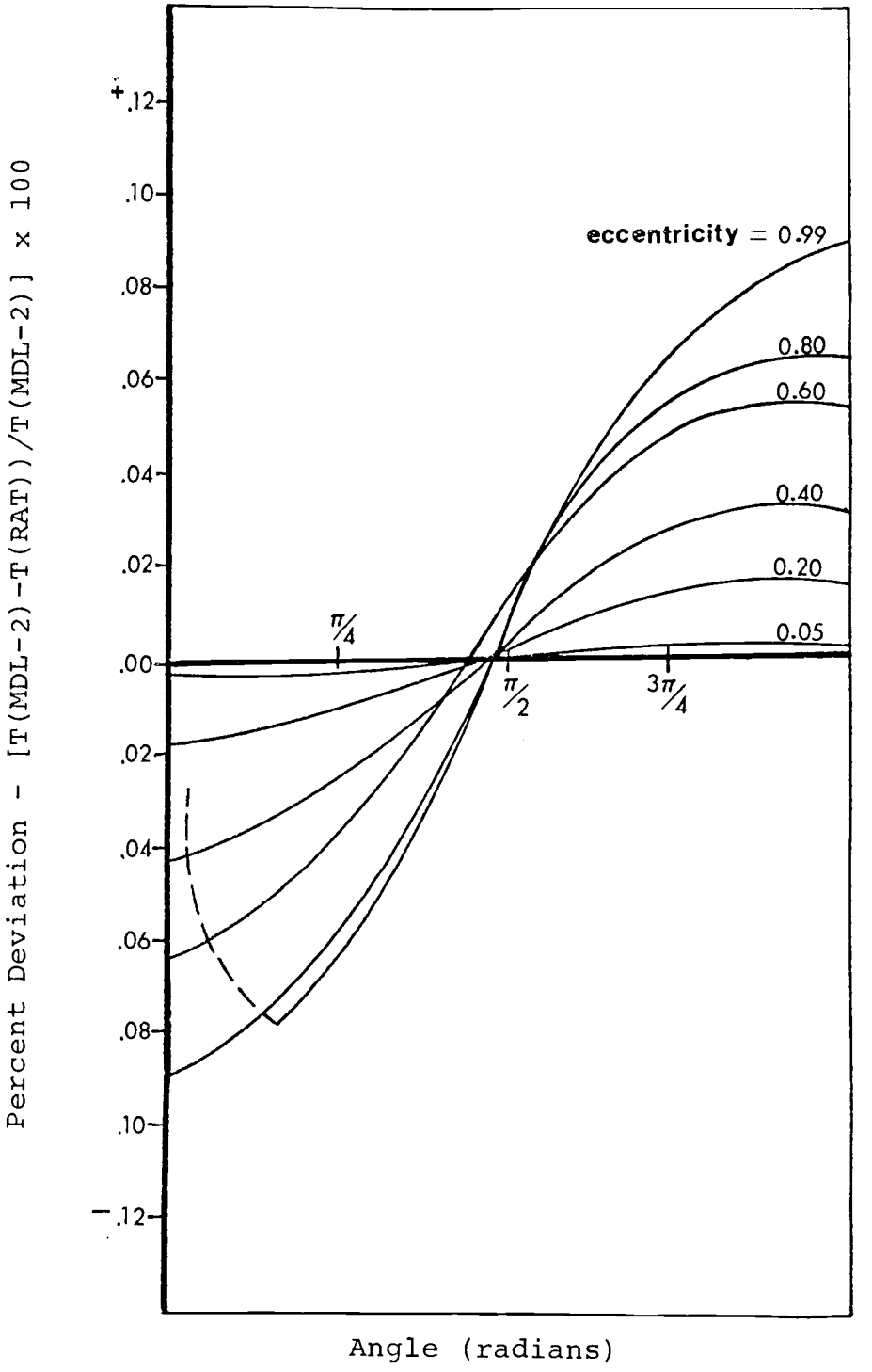


Figure 21. Percent deviation in maximum cladding temperatures (gap-width-to-fuel-diameter = 0.0676).

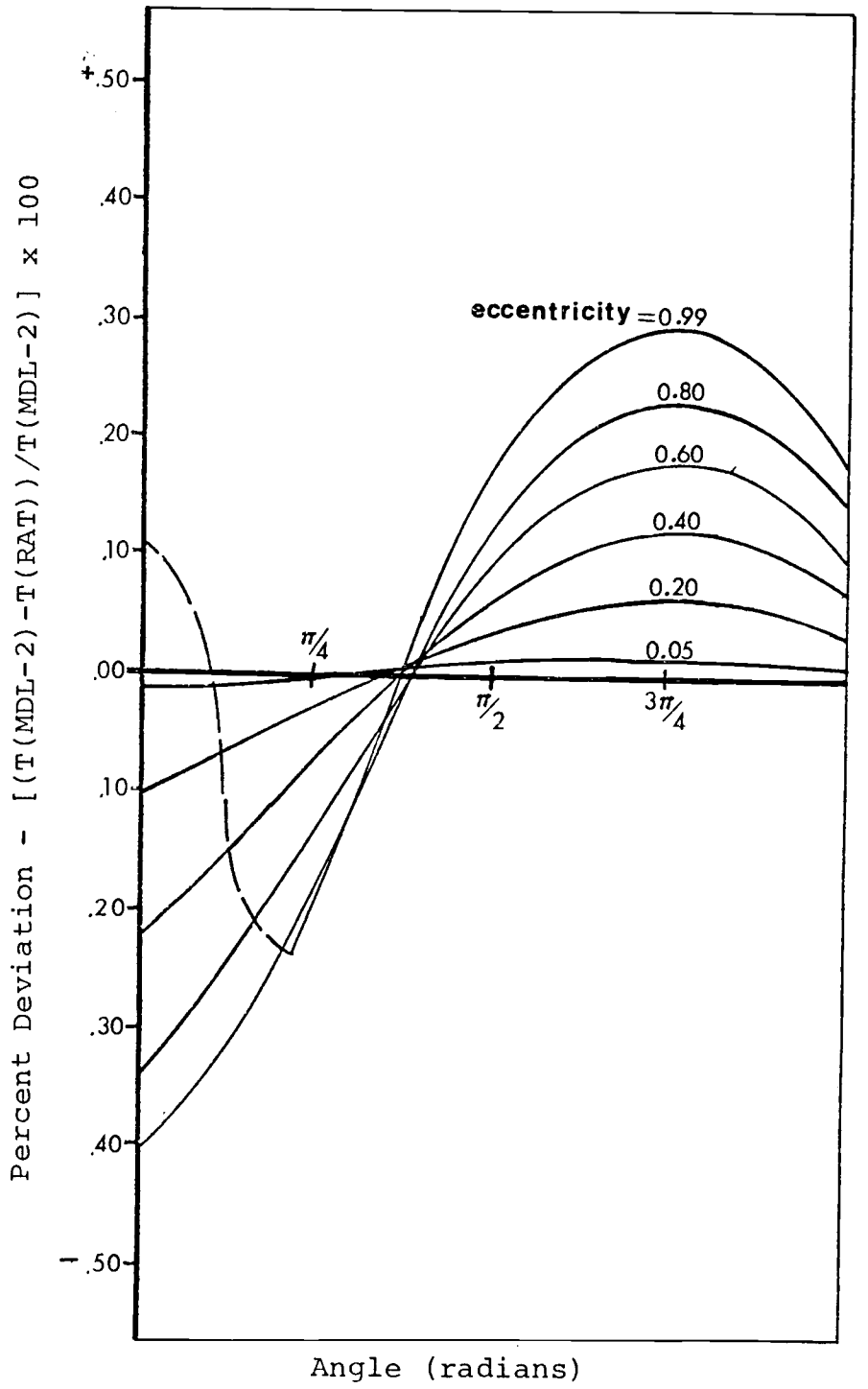
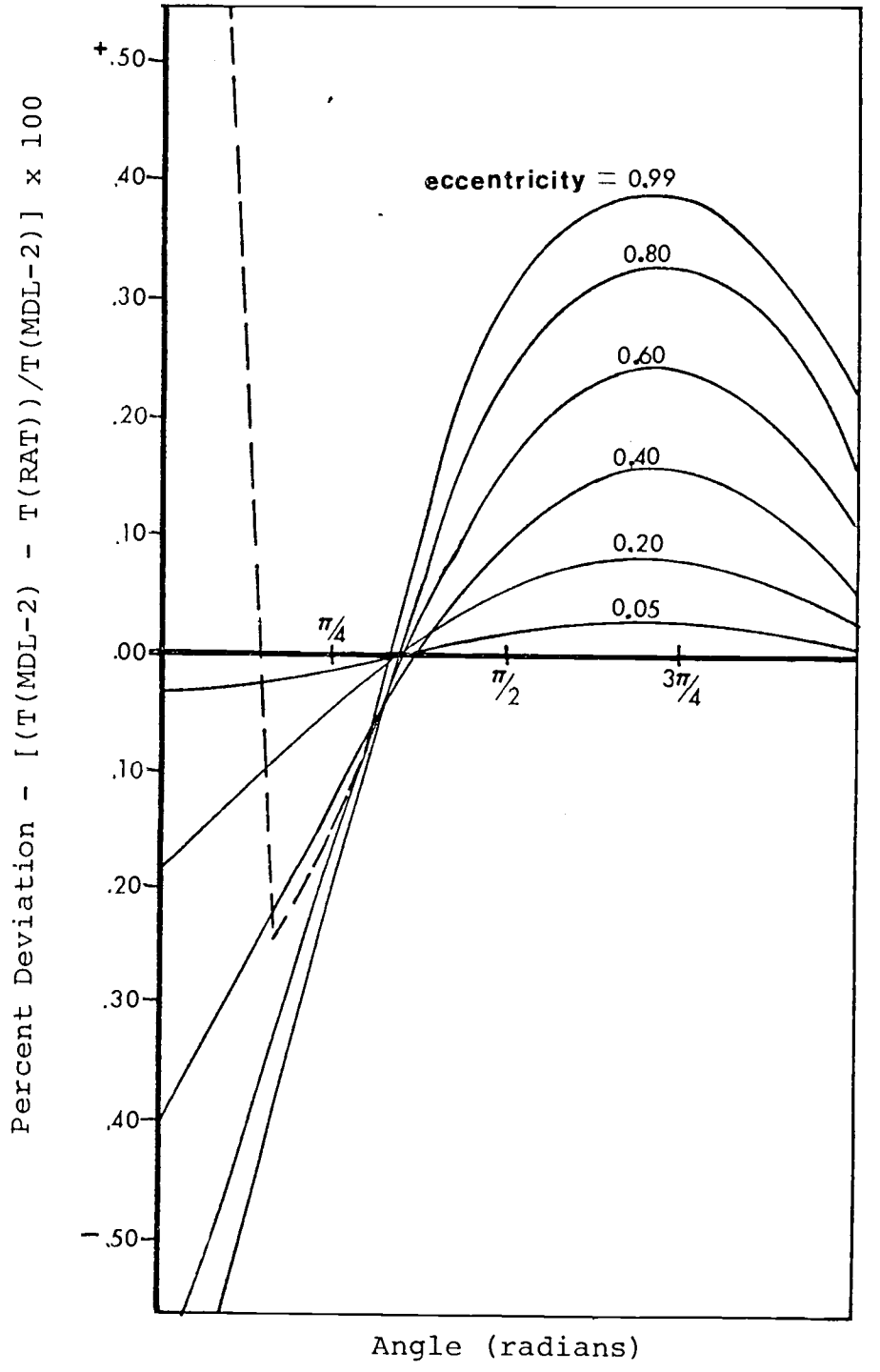


Figure 22. Percent deviation in maximum cladding temperatures (gap-width-to-fuel-diameter = 0.1563).



The lowered resistance within the gas gap for the improved modeled conductivity model has, in some cases, resulted in the temperatures predicted by the modeled approach being lower than those predicted by the ratchet boundary model.

5.3 Economics

The improved modeled conductivity model requires the same amount of iterations to converge as the original modeled conductivity model. This results in the ratchet boundary model requiring from 2 to 13 times as many iterations to converge than either of the other two models. Taking into account that each iteration of the ratchet boundary model requires approximately 3 times the amount of CPU time as the modeled conductivity approaches, results in the running time of the ratchet boundary model being from 6 to 40 times as long as either of the modeled conductivity approaches.

VI. CONCLUSIONS

The results of the comparison tests indicate that all three models represent an effective means of calculating cladding temperatures for most cases. The few cases that employ a very high eccentricity have shown the pellet surface averaging by the ratchet boundary method to under predict the temperatures within the cladding nearest the fuel pellet. In most cases the deviations in the maximum cladding temperatures, as predicted by the three models, was seen to be very small.

The main difference between the three models lies in the resistance to heat flow that is imposed within the gas gap. The ratchet boundary model and the improved modeled conductivity model make an attempt to describe the total heat flux vector leaving the fuel pellet while the original modeled conductivity model takes only the radial component into account. The ratchet boundary model and the improved modeled conductivity model can therefore be expected to generate lower resistances within the gas gap and lower temperatures within the fuel pellet than the original modeled conductivity model. As the comparison test indicate, this is indeed the case.

Until such a time that an analytical solution is available to provide a bench-mark for comparison with the

numerical solutions, it will be impossible to determine the method which generates temperature profiles most similar to the true solution.

At the present time, due to the small deviation between the temperatures predicted by the ratchet boundary model and the improved modeled conductivity model and the large difference in the economics of the two models, the improved modeled conductivity model has a distinct marked advantage as the finite difference method best suited to predicting the temperatures within an eccentrically loaded fuel element.

BIBLIOGRAPHY

1. K. L. Peddicord, B. D. Ganapol, R. Henninger, "A consistent algorithm for the study of heat transfer in eccentric annuli," Trans. Am. Nuc. Soc., 22 (1975).

2. R. C. Durfee, C. W. Nestor, Jr., "ORTHIS, ORTHAT - two computer programs for solving two-dimensional steady-state and transient heat conduction problems," ORNL-TM-3324, Contract No. W-7405-eng-26.

APPENDICES

APPENDIX A

NUMERICAL TECHNIQUES FOR STEADY-STATE HEAT CONDUCTION (2)

The equation which describes steady-state heat conduction in an isotropic material may be written in vector notation as

$$\vec{\nabla} \cdot k \vec{\nabla} T + q = 0 \qquad \text{(A-1)}$$

where $\vec{\nabla}$ = gradient operator

T = temperature

k = thermal conductivity

q = heat-generation rate per unit volume

Integrating the above equation over a nodal volume, V_k, results in

$$\int_{V_k} \vec{\nabla} \cdot k \vec{\nabla} T \, dV + \int_{V_k} q \, dV = 0 \qquad \text{(A-2)}$$

Using Green's first identity the equation may be rewritten as:

$$\int_{A_k} k \, \partial T/\partial N \, dA + \int_{V_k} q \, dV = 0 \qquad \text{(A-3)}$$

where the integration now takes place over the nodal surface, A_k. The $\partial T/\partial N$ represents the partial derivative of T

with respect to the outward normal to the surface of the k^{th} node. Assuming that k does not vary as we integrate over the surface area, and assuming that q does not vary as we integrate over the nodal volume, we obtain:

$$k_k \int_{A_k} \partial T/\partial N \, dA + q_k \, V_k = 0 \qquad \text{(A-4)}$$

A thin portion of one of the nodes in the mesh is shown in Figure 23. The four surfaces over which the surface integral is taken are numbered as shown. Assuming that $\partial T/\partial N$ does not vary over each surface of integration, equation (A-4) may be rewritten as:

$$k_k \sum_{S=1}^{4} A_{k,s} \frac{\partial T}{\partial N_{k,S}} + q_k \, V_k = 0 \qquad \text{(A-5)}$$

where $A_{k,s}$ = the area of side S, for node k

The evaluation of $\partial T/\partial N_{k,S}$ on each of the four sides will depend upon whether a given side is touching (1) a node in the same region, (2) a node in a different region, or (3) the outside boundary. In all cases, the $\partial T/\partial N_{k,S}$ may be approximated by an equation of the form

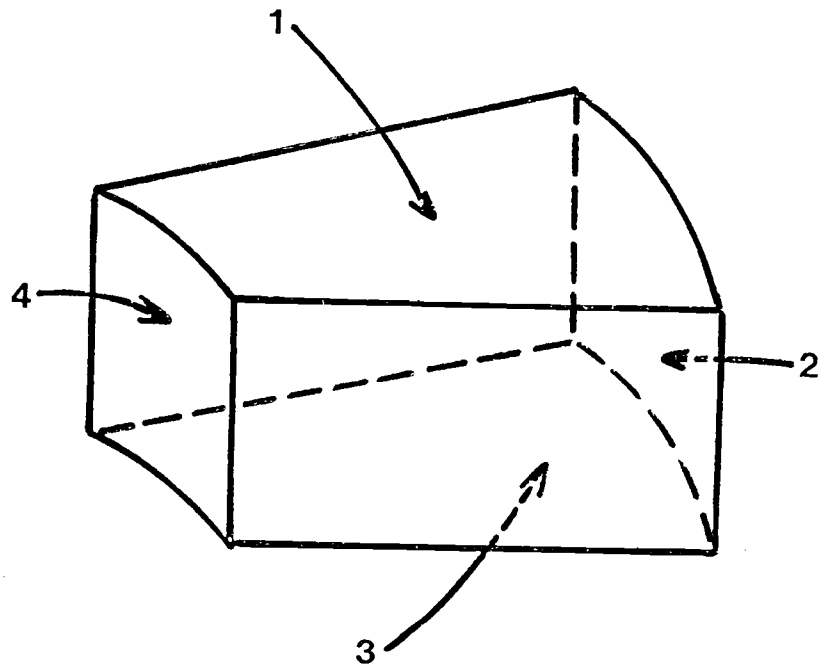
$$\frac{\partial T}{\partial N_{k,S}} = \frac{\tau_{k,S} - T_k}{B_{k,S}} \qquad \text{(A-6)}$$

where $\tau_{k,S}$ = neighboring temperature

T_k = temperature of the node itself

$B_{k,S}$ = internodal distance.

Figure 23. Surface areas for typical nodal volume.



$\tau_{k,S}$ and $B_{k,S}$ depend upon the type of neighboring node (or boundary) touching each side of the given node, k. $\tau_{k,S}$ and $B_{k,S}$ for the three cases given above are shown in Figure 24 for the side, S = 2.

Substituting equation (A-6) into equation (A-5) results in:

$$k_k \sum_{S=1}^{4} A_{k,S} \left(\frac{\tau_{k,S} - T_k}{B_{k,S}}\right) + q_k V_k = 0 \qquad \text{(A-7)}$$

Solving equation (A-7) for T_k results in the final equation for the temperature at node, k.

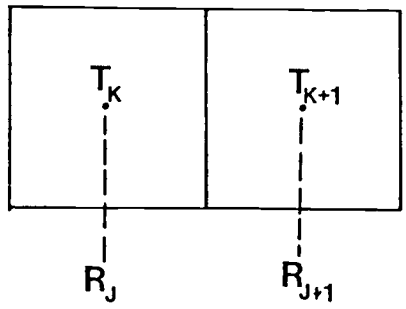
$$T_k = \frac{q_k V_k + k_k \sum_{S=1}^{4} \frac{A_{k,S} \tau_{k,S}}{B_{k,S}}}{k_k \sum_{S=1}^{4} \frac{A_{k,S}}{B_{k,S}}} \qquad \text{(A-8)}$$

The temperature of node k is expressed in terms of $\tau_{k,S}$, the temperature of its four surrounding neighbors. An initial guess for the temperatures at each node is made at the beginning of the problem. An iterative solution is then carried out in which new temperatures are calculated from old temperatures using equation (A-8).

An "overrelaxation" of the temperatures is often desirable to increase the convergence of the iterative procedure. This is done by using the constant, β, in the following manner:

Figure 24. Calculation of $\tau_{k,S}$ and $B_{k,S}$ for three given cases.

1.

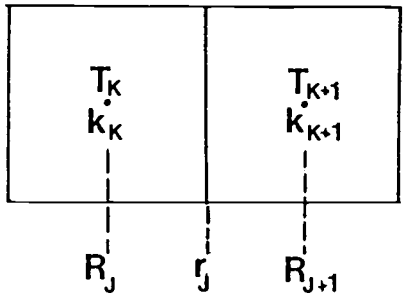


Neighbor = Node in same region

$$B_{K,2} = R_{J+1} - R_J$$

$$\tau_{K,2} = T_{K+1}$$

2.

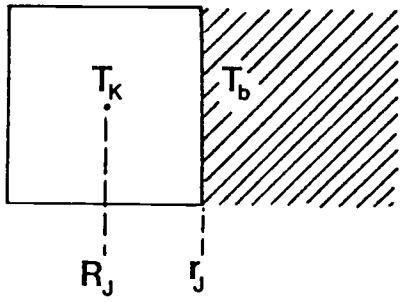


Neighbor = Node in different region

$$B_{K,2} = (r_J - R_J) + (R_{J+1} - r_J)\frac{k_K}{k_{K+1}}$$

$$\tau_{K,2} = T_{K+1}$$

3.



Neighbor = Outside boundary with fixed boundary temperature, T_b given

$$B_{K,2} = r_J - R_J$$

$$\tau_{K,2} = T_b$$

$$T_k^{New} = T_k^N + (1 - \beta) T_k^{N-1} \qquad \text{(A-9)}$$

where T_k^N = temperatures calculated during the Nth iteration using equation (A-8)

T_k^{N-1} = final "overrelaxed, extrapolated" temperatures resulting from the (N-1)th iteration

T_k^{New} = the "overrelaxed" temperatures being calculated during this Nth iteration

β = input factor between 1 and 2

After the overrelaxation is finished, another acceleration technique, "dominant-error-mode extrapolation", is applied. At each iteration an error term is calculated for the mesh as a whole.

$$e_N = \sum_{k=1}^{k_{max}} \left| \frac{T_k^{N-1}}{T_k^N} - 1 \right| \qquad \text{(A-10)}$$

A ratio of these error terms gives an extrapolation factor, f,

$$f = \frac{e_N}{e_{N-1}} \qquad \text{(A-11)}$$

which is used to calculate a better estimate of the temperatures.

$$T_k^{final} = T_k^{New} + \frac{f}{1-f} (T_k^{New} - T_k^{N-1}) \qquad \text{(A-12)}$$

T_k^{New} and T_k^{N-1} are defined above and T_k^{final} is now referred to as the final "overrelaxed, extrapolated" temperature, T_k^N, for the Nth iteration. This extrapolation is not done if the factor, f, in equation (A-11) is greater than one. In actual practice, this extrapolation should only be used every 40th or 50th iteration, depending on the problem, so that the temperatures have a chance to settle down before being kicked again. The frequency of extrapolation is an input quantity which is used for the whole machine run.

Successive iterations are carried out until

$$\underset{k}{MAX} \left| \frac{T_k^{N-1}}{T_k^N} - 1 \right| \leq \varepsilon \qquad \text{(A-13)}$$

where ε = specified convergence criterion

The error term is calculated for every node in the mesh, but only the maximum is used in equation (A-13).

APPENDIX B

COMPUTER PROGRAMS

B.1 RAT, MDL, and MDL-2

RAT, MDL, and MDL-2 are finite difference programs developed to calculate the temperatures generated within an eccentrically loaded fuel pin. All three codes use the equations and techniques derived in Appendix A for a standard steady-state heat conduction problem. The difference in the three programs is found in their treatment of the gas gap region.

RAT assigns a constant conductivity for the gas gap region and accounts for the eccentricity by a ratchet boundary on the fuel pellet surface. The ratchet boundary acts to average the distance between the fuel pellet and the cladding for each angular region. The radius of each of the ratchet boundary arcs is given by equation 2.4.

$$\rho = d \cos \phi + [R^2 - d^2(1 - \cos^2 \phi)]^{\frac{1}{2}} \qquad 2.4$$

MDL accounts for the eccentricity of the problem by defining a geometrically dependent variable conductivity within the gas gap. The conductivity within the gas gap is given by equation 2.21.

$$k(\phi) = k \frac{\Delta R}{G(\phi)} \qquad 2.21$$

MDL-2 uses the same method as MDL, but makes a better approximation of the geometrically dependent conductivity within the gas gap. The conductivity within the gas gap is now given by equation 4.9.

$$k(\phi) = k[\cos(\theta-\phi) + \sin(\theta-\phi)] \frac{\Delta R}{G(\phi)} \qquad 4.9$$

All three codes use tape 5 for an input. A detailed description of the input requirements is presented here and followed by a listing of the programs.

CARD 1: NTHREG FORMAT(1X,I4)
NTHREG - THE NUMBER OF ANGULAR REGIONS

CARD 2: NDFUEL FORMAT(1X,I4)
NDFUEL - THE NUMBER OF RADIAL NODE REGIONS IN THE FUEL

CARD 3: NDCLAD FORMAT(1X,I4)
NDCLAD - THE NUMBER OF RADIAL NODE REGIONS IN THE CLAD

CARD 4: BETA FORMAT(1X, F12.8)
BETA - THE SUCCESSIVE OVER-RELAXATION FACTOR (OPTIMUM = 1.84)

CARD 5: EPS FORMAT(1X, F12.8)
EPS - THE CONVERGENCE CRITERION FOR TEMPERATURES

CARD 6: CONDCLA FORMAT(1X, F12.8)
CONDCLA - THE CONDUCTIVITY OF THE CLADDING (W/cm°C)

CARD 7: CONDGAS FORMAT(1X, F12.8)
CONDGAS - THE CONDUCTIVITY OF THE GAS WITHIN THE GAS GAP (W/cm°C)

CARD 8: TEMP1ST FORMAT(1X, F12.8)
TEMP1ST - INITIAL GUESS AT NODAL TEMPERATURES (°C)

CARD 9: TEMPBLK FORMAT(1X, F12.8)
TEMPBLK - OUTSIDE CLADDING TEMPERATURE (°C)

CARD 10: NALLOWD FORMAT(1X, I4)
NALLOWD - MAXIMUM NUMBER OF ITERATIONS ALLOWED

CARD 11: RCLAD FORMAT(1X, F12.8)
RCLAD - OUTSIDE RADIUS OF CLAD (cm)

CARD 12: CONDFUE FORMAT(1X, F12.8)
CONDFUE - CONDUCTIVITY OF FUEL MATERIAL ($W/cm^{\circ}C$)

CARD 13: SHIFT FORMAT(1X, F12.8)
SHIFT - ECCENTRICITY OF PROBLEM AS A FRACTION

CARD 14: RFUELA FORMAT(1X, F12.8)
RFUELA - RADIUS OF FUEL PELLET (cm)

CARD 15: GASWDTH FORMAT(1X, F12.8)
GASWDTH - GAS GAP WIDTH (cm)

CARD 16: SOURCE FORMAT(1X, F12.8)
SOURCE - HEAT GENERATION RATE FOR FUEL (W/cm^3)

PROGRAM LISTING FOR RAT

```
      PROGRAM PAT (INPUT,OUTPUT,TAPE5,TAPE6)
      DIMENSION AREA(40,40,4), BS(40,40,4)
      DIMENSION VOLUME(40,40),TEMP(40,40), OLDTEMP(40,40)
      DIMENSION RADLIN(50),RADNOD(40),THELIN(50),THENOD(40)
      DIMENSION TDIFF(2)
      DIMENSION COND(40,40),QK(40,40)
      COMMON/HEATSRC/SOURCE
      COMMON/TEMPOLD/ OLDTEMP(40,40)
      COMMON/TEMP/ TEMP(40,40),TEMPBLK,AREA(40,40,4)
      COMMON/BETA/ BS(40,40,4),QK(40,40),VOLUME(40,40),BETA
      COMMON/NODPNTS/NOFUEL,INNODES,NODEND,NTHREG
      COMMON/CONDUCT/COND(40,40)
      COMMON/SPACING/RADNOD(40),RADLIN(50),THESPAC
      PI=3.14159265536
C
C          INPUT
C
      READ(5,2000) NTHREG
      READ(5,2000) NOFUEL
      READ(5,2000) NOCLAD
      READ(5,2001) BETA
      READ(5,2001) EPS
      READ(5,2001) CONDCLA
      READ(5,2001) CONDGAS
      READ(5,2001) TEMP1ST
      READ(5,2001) TEMPBLK
      READ(5,2000) NALLOWD
      READ(5,2001) RCLAD
      READ(5,2001) CONDFUE
      READ(5,2001) SHIFT
      READ(5,2001) RFUELA
      READ(5,2001) GASWTH
      READ(5,2001) SOURCE
 2000 FORMAT(1X,I4)
 2001 FORMAT(1X,F12.8)
C
C          DETERMINATION OF NODAL LINE SPACINGS
C
      INNODES=NOFUEL+NTHREG+1
```

```
      LINEND=NOFUEL+NOCLAD+NTHREG+2
      NODEND=NOFUEL+NOCLAD+NTHREG+1
      THESPAC=PI/NTHREG
      DIFFER=SHIFT*GASWDTH
      RFUELO=RFUELA-DIFFER
      AREAFO=PI*RFUELO*RFUELO/NOFUEL
      RINCLAD=RFUELA+GASWDTH
      AREACA=(RCLAD*RCLAD-RINCLAD*RINCLAD)*PI/NOCLAD
      RADLIN(1)=0.
      DO 1 J=2,LINEND
      IF(J.GT.NOFUEL+1.AND.J.LE.NOFUEL+NTHREG+1) GO TO 2
      IF(J.GT.INNODES) GO TO 3
      RADLIN(J)=SQRT((J-1)*AREAFO/PI)
      GO TO 4
 2    ANGLE=(J-NOFUEL-1)*THESPAC-THESPAC/2.
      RADLIN(J)=(SQRT((2*DIFFER*COS(ANGLE))**2+4*(RFUELA*
     vRFUELA-DIFFER*DIFFER))-2*DIFFER*COS(ANGLE))/2.
      GO TO 4
 3    RADLIN(J)=SQRT(RINCLAD**2+(J-INNODES-1)*AREACA/PI)
 4    RADNOD(J-1)=(RADLIN(J)+RADLIN(J-1))/2.
 1    CONTINUE
      THELIN(1)=0.
      DO 5 J=1,NTHREG
      THELIN(J+1)=THELIN(J)+THESPAC
      THENOD(J)=(THELIN(J)+THELIN(J+1))/2.
 5    CONTINUE
C
C          INITIALIZE NODAL TEMPERATURES
C          AND ASSIGN ≠0≠ VALUES
C
      DO 50 I=1,NTHREG
      DO 50 J=1,NODEND
      TEMP(I,J)=TEMP1ST
      QK(I,J)=0.
      IF(J.LE.NOFUEL+1) QK(I,J)=SOURCE
C
C          ASSIGN CONDUCTIVITY VALUES
C
      COND(I,J)=CONDFUE
```

```
      IF(J.GT.NOFUEL+I.AND.J.LE.INNODES) COND(I,J)=CONDGAS
      IF(J.GT.INNODES) COND(I,J)=CONDCLA
 50   CONTINUE
C
C          CALCULATE NODAL VOLUMES AND SURFACE AREAS
C
      DO 6 I=1,NTHREG
      DO 6 J=1,NODEND
      VOLUME(I,J)=(RADLIN(J+1)**2-RADLIN(J)**2)*THESPAC/2.
      DO 6 K=1,4
      GO TO (7,8,9,10),K
 7    AREA(I,J,K)=RADLIN(J+1)*THESPAC
      GO TO 12
 8    IF(I.EQ.1) GO TO 11
      AREA(I,J,K)=RADLIN(J+1)-RADLIN(J)
      GO TO 12
 9    AREA(I,J,K)=RADLIN(J)*THESPAC
      GO TO 12
 10   IF(I.EQ.NTHREG) GO TO 11
      AREA(I,J,K)=RADLIN(J+1)-RADLIN(J)
      GO TO 12
 11   AREA(I,J,K) =0.
C
C          CALCULATE INTERNODAL DISTANCES
C
 12   RS(I,J,K)=GETRS(I,J,K)
 6    CONTINUE
C
C          INITIALIZE ITERATION COUNTERS
C
      NEWCNT=100
      NSTOP=0
      ITCOUNT=0
 29   NSPEED=0
 23   IF( ITCOUNT.GT. NALLOWD) GO TO 30
      IF(NSPEED.GE. 49) GO TO 25
C
C          ITERATION CALCULATION OF TEMPERATURES
C
```

```
 27     DO 21 J=1,NODENO
        DO 21 I=1,NTHREG
        OLDTEMP(I,J)=TEMP(I,J)
        TEMP(I,J)=TEMPNEW(I,J)
 21     CONTINUE
C
C           CHECK FOR CONVERGENCE
C
 25     DO 24 I=1,NTHREG
        DO 24 J=1,NODENO
        IF(NEWCNT.LT.100) GO TO 22
        IF(ABS(TEMP(I,J)/OLDTEMP(I,J)-1).GT.EPS) GO TO 22
        GO TO 24
 22     ITCOUNT=ITCOUNT+1
        NEWCNT=NEWCNT+1
        NSPEED=NSPEED+1
        GO TO 23
 24     CONTINUE
        GO TO 30
C
C           DOMINANT-ERROR-MODE ACCELERATION
C
 25     TDSUM=0.
        DO 26 I=1,NTHREG
        DO 26 J=1,NODENO
        TRATIO=ABS(OLDTEMP(I,J)/TEMP(I,J)-1.)
        TDSUM=TDSUM+TRATIO
 26     CONTINUE
        TDIFF(NSPEED-48)=TDSUM
        IF(NSPEED.NE.50) GO TO 27
        ARATIO=TDIFF(2)/TDIFF(1)
        ALPHA=0.0
        IF(ARATIO.NE.1.0) ALPHA=1.0/(1.-ARATIO)
        IF(ARATIO.GE.1.0) GO TO 29
        DO 28 I=1,NTHREG
        DO 28 J=1,NODENO
 28     TEMP(I,J)=ALPHA*TEMP(I,J)+(1.-ALPHA)*OLDTEMP(I,J)
        GO TO 29
C
```

```
C          OUTPUT
C
 30    WRITE(6,40) RFUELA,RCLAD,GASWDTH,SHIFT,SOURCE,CONDFUE
     v,CONDCLA,CONDGAS,EPS
 40    FORMAT(62X,*INPUT DATA*,//,55X,*RADIUS OF FUEL=*,F10.7
     v,/,55X,*RADIUS OF CLAD=*,F10.7,/,55X,*GAS WIDTH=*,F10.
     v7,/,55X,*FUEL SHIFT=*,F10.7,/,55X,*HEAT GENERATION=*,
     vF12.7,/,55X,*CONDUCTIVITY OF FUEL=*,F10.7,/.55X,
     v*CONDUCTIVITY OF CLAD=*,F10.7,/,55X,
     v*CONDUCTIVITY OF GAS=*,F10.7,/,55X,*EPSILON=*,F10.7,
     v///)
       WRITE(6,41) ITCOUNT
 41    FORMAT(45X,*FINAL TEMPERATURE AFTER*,2X,I4,2X,
     v*ITERATIONS*,/)
       NPREV=1
       NCOUNT=10
 31    WRITE(6,42)
 42    FORMAT(5X,126(*=*) )
       DO 33 I=1,NTHREG
       IF(NCOUNT.GT.NODEND) NCOUNT=NODEND
       WRITE(6,32) (TEMP(I,J), J=NPREV,NCOUNT)
 32    FORMAT(5X,*=*,2X,10(F8.3,4X),2X,*=* )
 33    CONTINUE
       WRITE(6,42)
       WRITE(6,43) (RADNOD(J),J=NPREV,NCOUNT)
 43    FORMAT(1X,*SPACING*,10(F8.3,4X),//)
       NPREV=NCOUNT+1
       NCOUNT=NCOUNT+10
       IF(NCOUNT-NODEND.LT.10) GO TO 31
       NEWCNT=0
       NSTOP=NSTOP+1
       IF(NSTOP.EQ.1) GO TO 23
       END
       FUNCTION GETBS(L,M,N)
C
C          FUNCTION TO DETERMINE INTERNODAL DISTANCES FOR
C          EACH NODE TO EACH OF 4 NEIGHBORS (OR BOUNDARY)
C
       COMMON/CONDUCT/CCND(40,40)
```

```
      COMMON/NODPNTS/NOFUEL,INNODES,NODEND,NTHREG
      COMMON/SPACING/RADNOD(40),RADLIN(50),THESPAC
      GO TO (1,2,3,4),N
C
C          OUTSIDE FACE OF NODE
C
 1    IF(M.EQ.NOFUEL+L) GO TO 20
      IF(M.EQ.INNODES) GO TO 20
      IF(M.EQ.NODEND) GO TO 30
      GETRS=RADNOD(M+1)-RADNOD(M)
      RETURN
C
C          RIGHT SIDE FACE OF NODE
C
 2    IF(L.EQ.1) GO TO 50
      IF(M.EQ.NOFUEL+L) GO TO 40
      GETRS=RADNOD(M)*THESPAC
      RETURN
C
C          INSIDE FACE OF NODE
C
 3    IF(M.EQ.1) GO TO 50
      IF(M.EQ.NOFUEL+L+1) GO TO 60
      IF(M.EQ.INNODES+1) GO TO 60
      GETRS=RADNOD(M)-RADNOD(M-1)
      RETURN
C
C          LEFT SIDE FACE OF NODE
C
 4    IF(L.EQ. NTHREG) GO TO 50
      IF(M.EQ.NOFUEL+L+1) GO TO 70
      GETRS=RADNOD(M)*THESPAC
      RETURN
 20   GETRS=RADLIN(M+1)-RADNOD(M)+(RADNOD(M+1)-RADLIN(M+1))
     $*COND(L,M)/COND(L,M+1)
      RETURN
 30   GETRS=RADLIN(M+1)-RADNOD(M)
      RETURN
 40   GETRS=RADNOD(M)*THESPAC/2.*(1.+COND(L,M)/COND(L-1,M))
```

```
      RETURN
   50 GETBS=1.
      RETURN
   60 GETBS=RADNOD(M)-RADLIN(M)+(RADLIN(M)-RADNOD(M-1))
     *COND(L,M)/COND(L,M-1)
      RETURN
   70 GETBS=RADNOD(M)*THESPAC/2.*(1.+COND(L,M)/COND(L+1,M))
      RETURN
      END
      FUNCTION TEMPNEW(IDUM,JDUM)
C
C          FUNCTION TO CALCULATE NEW NODAL TEMPERATURES
C
      COMMON/TEMP/ TEMP(40,40),TEMPBLK,AREA(40,40,4)
      COMMON/CONDUCT/ COND(40,40)
      COMMON/TEMPOLD/ OLDTEMP(40,40)
      COMMON/BETA/ BS(40,40,4),QK(40,40),VOLUME(40,40),BETA
      COMMON/NODPNTS/ NOFUEL,INNODES,NODEND,NTHREG
      SUMTOP=0.
      SUMBOT=0.
C
C          SELECT TEMPERATURE OF NEIGHBORING NODE
C
      DO 20 K=1,4
      GO TO(14,15,16,17) ,K
   14 IF(JDUM.EQ.NODEND) GO TO 18
      TEMPNBR=TEMP(IDUM,JDUM+1)
      GO TO 19
   15 IF(IDUM.EQ.1) GO TO 20
      TEMPNBR=TEMP(IDUM-1,JDUM)
      GO TO 19
   16 IF(JDUM.EQ.1) GO TO 20
      TEMPNBR=TEMP(IDUM,JDUM-1)
      GO TO 19
   17 IF(IDUM.EQ.NTHREG) GO TO 20
      TEMPNBR=TEMP(IDUM+1,JDUM)
      GO TO 19
   18 TEMPNBR=TEMPBLK
   19 CONTINUE
```

```
      SUMTOP=SUMTOP+COND(IDUM,JDUM)*AREA(IDUM,JDUM,K)*
     $TEMPNBR/BS(IDUM,JDUM,K)
      SUMBOT=SUMBOT+COND(IDUM,JDUM)*AREA(IDUM,JDUM,K)/
     vBS(IDUM,JDUM,K)
 20   CONTINUE
C
C          CALCULATION OF TEMPERATURE
C
      TEMP(IDUM,JDUM)=(QK(IDUM,JDUM)*VOLUME(IDUM,JDUM)
     v+SUMTOP)/SUMBOT
C
C          OVERRELAXATION OF TEMPERATURES
C
      TEMP(IDUM,JDUM)=BETA*TEMP(IDUM,JDUM)+(1-BETA)*
     vOLDTEMP(IDUM,JDUM)
      TEMPNEW=TEMP(IDUM,JDUM)
      RETURN
      END
```

PROGRAM LISTING FOR MDL

```
      PROGRAM MOL (INPUT,OUTPUT,TAPE5,TAPE6)
      DIMENSION AREA(40,40,4), BS(40,40,4)
      DIMENSION VOLUME(40,40),TEMP(40,40), OLDTEMP(40,40)
      DIMENSION RADLIN(50),RADNOD(40),THELIN(50),THENOD(40)
      DIMENSION TDIFF(2)
      DIMENSION COND(40,40),QK(40,40)
      COMMON/HEATSRC/SOURCE
      COMMON/TEMPOLD/ OLDTEMP(40,40)
      COMMON/TEMP/ TEMP(40,40),TEMPBLK,AREA(40,40,4)
      COMMON/BETA/ BS(40,40,4),QK(40,40),VOLUME(40,40),BETA
      COMMON/NODPNTS/NOFUEL,INNODES,NODEND,NTHREG
      COMMON/CONDUCT/COND(40,40)
      COMMON/SPACING/RADNOD(40),RADLIN(50),THESPAC
      PI=3.14159265536
C
C          INPUT
C
      READ(5,2000) NTHREG
      READ(5,2000) NOFUEL
      READ(5,2000) NOCLAD
      READ(5,2001) BETA
      READ(5,2001) EPS
      READ(5,2001) CONDCLA
      READ(5,2001) CONDGAS
      READ(5,2001) TEMP1ST
      READ(5,2001) TEMPBLK
      READ(5,2000) NALLOWD
      READ(5,2001) RCLAD
      READ(5,2001) CONDFUE
      READ(5,2001) SHIFT
      READ(5,2001) RFUELA
      READ(5,2001) GASWDTH
      READ(5,2001) SOURCE
 2000 FORMAT(1X,I4)
 2001 FORMAT(1X,F12.8)
C
C          DETERMINATION OF NODAL LINE SPACINGS
C
      INNODES=NOFUEL+1
```

```
      LINEND=NDFUEL+NDCLAD+2
      NODEND=LINEND-1
      THESPAC=PI/NTHREG
      DIFFER=SHIFT*GASWDTH
      RFUELD=RFUELA-DIFFER
      AREAFD=PI*RFUELD*RFUELD/NDFUEL
      RINCLAD=RFUELA+GASWDTH
      AREACA=(RCLAD*RCLAD-RINCLAD*RINCLAD)*PI/NDCLAD
      RADLIN(1)=0.
      DO 1 J=2,LINEND
      IF(J.EQ. INNODES+1) GO TO 2
      IF(J.GT. INNODES+1) GO TO 3
      RADLIN(J)=SQRT((J-1)*AREAFD/PI)
      GO TO 4
 2    RADLIN(J)=RFUELA+GASWDTH
      GO TO 4
 3    RADLIN(J)=SQRT((RFUELA+GASWDTH)**2+(J-INNODES-1)*
     vAREACA/PI)
 4    RADNOD(J-1)=(RADLIN(J)+RADLIN(J-1))/2.
 1    CONTINUE
      THELIN(1)=0.
      DO 5 J=1,NTHREG
      THELIN(J+1)=THELIN(J)+THESPAC
      THENOD(J)=(THELIN(J)+THELIN(J+1))/2.
 5    CONTINUE
C
C          INITIALIZE NODAL TEMPERATURES
C          AND ASSIGN ≠G≠ VALUES
C
      DO 50 I=1,NTHREG
      DO 50 J=1,NODEND
      VOLUME(I,J)=(RADLIN(J+1)**2-RADLIN(J)**2)*THESPAC/2.
      TEMP(I,J)=TEMP1ST
      QK(I,J)=0.
      IF(J.LE.NDFUEL  ) QK(I,J)=SOURCE
C
C          ASSIGN CONDUCTIVITY VALUES
C
      COND(I,J)=CONDFUE
```

```
      IF(J.EQ.INNODES) GO TO 6000
      IF(J.GT.INNODES) COND(I,J)=CONDCLA
      GO TO 50
C
C          MODELED CONDUCTIVITY AS A FUNCTION OF ANGLE
C
 6000 IF(SHIFT.EQ.0.0) GO TO 6001
      PHI=THENOD(I)
      CS=COS(PHI)
      R=RFUELA+GASWDTH
      G=R-(DIFFER*CS+SQRT(RFUELA**2-DIFFER**2*(1-CS**2)))
      COND(I,J)=CONDGAS*GASWDTH/G
      GO TO 50
 6001 COND(I,J)=CONDGAS
 50   CONTINUE
      DO 6 I=1,NTHREG
      DO 6 J=1,NODEND
C
C          CALCULATE NODAL VOLUMES AND SURFACE AREAS
C
      VOLUME(I,J)=(RADLIN(J+1)**2-RADLIN(J)**2)*THESPAC/2.
      DO 6 K=1,4
      GO TO (7,8,9,10),K
 7    AREA(I,J,K)=RADLIN(J+1)*THESPAC
      GO TO 12
 8    IF(I.EQ.1) GO TO 11
      AREA(I,J,K)=RADLIN(J+1)-RADLIN(J)
      GO TO 12
 9    AREA(I,J,K)=RADLIN(J)*THESPAC
      GO TO 12
 10   IF(I.EQ.NTHREG) GO TO 11
      AREA(I,J,K)=RADLIN(J+1)-RADLIN(J)
      GO TO 12
 11   AREA(I,J,K) =0.
C
C          CALCULATE INTERNODAL DISTANCES
C
 12   ES(I,J,K)=GETDS(I,J,K)
 6    CONTINUE
```

```
C
C          INITIALIZE ITERATION COUNTERS
C
      ITCOUNT=0
 29   NSPEED=0
 23   IF( ITCOUNT.GT. NALLOWD) GO TO 30
      IF(NSPEED.GE. 49) GO TO 25
C
C          ITERATION CALCULATION OF TEMPERATURES
C
 27    DO 21  J=1,NODEND
      DO 21  I=1,NTHREG
      II=NTHREG+1-I
      OLDTEMP(II,J)=TEMP(II,J)
      TEMP(II,J)=TEMPNEW(II,J)
 21   CONTINUE
C
C          CHECK FOR CONVERGENCE
C
 75   DO 24  I=1,NTHREG
      DO 24  J=1,NODEND
      IF(ABS(TEMP(I,J)/OLDTEMP(I,J)-1).GT.EPS) GO TO 22
      GO TO 24
 22   ITCOUNT=ITCOUNT+1
      NSPEED=NSPEED+1
      GO TO 23
 24   CONTINUE
      GO TO 30
C
C          DOMINANT-ERROR-MODE ACCELERATION
C
 25   TDSUM=0.
      DO 26  I=1,NTHREG
      DO 26  J=1,NODEND
      TRATIO=ABS(OLDTEMP(I,J)/TEMP(I,J)-1.)
      TDSUM=TDSUM+TRATIO
 26   CONTINUE
      TDIFF(NSPEED-48)=TDSUM
      IF(NSPEED.NE.50) GO TO 27
```

```
      ARATIO=TDIFF(2)/TDIFF(1)
      ALPHA=0.0
      IF(ARATIO.NE.1.0) ALPHA=1.0/(1.-ARATIO)
      IF(ARATIO.GE.1.0) GO TO 29
      DO 28 I=1,NTHREG
      DO 28 J=1,NODEND
 28   TEMP(I,J)=ALPHA*TEMP(I,J)+(1.-ALPHA)*OLDTEMP(I,J)
      GO TO 29
C
C          OUTPUT
C
 30   WRITE(6,40)RFUEL,RCLAD,GASWDTH,SHIFT,SOURCE,CONDFUE,
     vCONDCLA,CONDGAS,EPS
 40   FORMAT(62X,'INPUT DATA',//,55X,'RADIUS OF FUEL=',F10.7
     v,/,55X,'RADIUS OF CLAD=',F10.7,/,55X,'GAS WIDTH=',F10.
     v7,/,55X,'FUEL SHIFT=',F10.7,/,55X,'HEAT GENERATION=',
     vF12.7,/,55X,'CONDUCTIVITY OF FUEL=',F10.7,/,55X,
     v'CONDUCTIVITY OF CLAD=',F10.7,/,55X,
     v'CONDUCTIVITY OF GAS=',F10.7,/,55X,'EPSILON=',F10.7,
     v///)
      WRITE(6,41) ITCOUNT
 41   FORMAT(45X,'FINAL TEMPERATURE AFTER',2X,I4,2X,
     v'ITERATIONS',/)
      NPREV=1
      NCOUNT=10
 31   WRITE(6,42)
      DO 33 I=1,NTHREG
      II=NTHREG+1-I
      IF(NCOUNT.GT.NODEND)NCOUNT=NODEND
      WRITE(6,32)(TEMP(II,J),J=NPREV,NCOUNT)
 32   FORMAT(5X,'*',2X,10(F8.3,4X),2X,'*',/)
 33   CONTINUE
      WRITE(6,42)
 42   FORMAT(5X,126('*') )
      WRITE(6,43)(RADNOD(J),J=NPREV,NCOUNT)
 43   FORMAT(1X,'SPACING',10(F8.3,4X),//)
      NPREV=NCOUNT+1
      NCOUNT=NCOUNT+10
      IF(NCOUNT-NODEND.LT.10) GO TO 31
```

```
      END
      FUNCTION GETBS(L,M,N)
C
C         FUNCTION TO DETERMINE INTERNODAL DISTANCES FOR
C         EACH NODE TO EACH OF 4 NEIGHBORS (OR BOUNDARY)
C
      COMMON/CONDUCT/CCND(40,40)
      COMMON/NODPNTS/NOFUEL,INNODES,NODEND,NTHREG
      COMMON/SPACING/RADNOD(40),RADLIN(50),THESPAC
      GO TO (1,2,3,4),N
C
C         OUTSIDE FACE OF NODE
C
 1    IF(M.EQ.NOFUEL  ) GO TO 20
      IF(M.EQ.INNODES) GO TO 20
      IF(M.EQ.NODEND) GO TO 30
      GETBS=RADNOD(M+1)-RADNOD(M)
      RETURN
C
C         RIGHT SIDE FACE OF NODE
C
 2    IF(L.EQ.1) GO TO 50
      GETBS=RADNOD(M)*THESPAC
      RETURN
C
C         INSIDE FACE OF NODE
C
 3    IF(M.EQ.1) GO TO 50
      IF(M.EQ.INNODES    ) GO TO 60
      IF(M.EQ.INNODES+1) GO TO 60
      GETBS=RADNOD(M)-RADNOD(M-1)
      RETURN
C
C         LEFT SIDE FACE OF NODE
C
 4    IF(L.EQ. NTHREG) GO TO 50
      GETBS=RADNOD(M)*THESPAC
      RETURN
 20   GETBS=RADLIN(M+1)-RADNOD(M)+(RADNOD(M+1)-RADLIN(M+1))
```

```
     $*COND(L,1)/COND(L,M+1)
      RETURN
   30 GETBS=RADLIN(M+1)-RADNOD(M)
      RETURN
   50 GETBS=1.
      RETURN
   60 GETBS=RADNOD(M)-RADLIN(M)+(RADLIN(M)-RADNOD(M-1))
     $*COND(L,M)/COND(L,M-1)
      RETURN
      END
      FUNCTION TEMPNEW(IDUM,JDUM)
C
C         FUNCTION TO CALCULATE NEW NODAL TEMPERATURES
C
      COMMON/TEMP/ TEMP(40,40),TEMPBLK,AREA(40,40,4)
      COMMON/CONDUCT/ COND(40,40)
      COMMON/TEMPOLD/ OLDTEMP(40,40)
      COMMON/BETA/ BS(40,40,4),QK(40,40),VOLUME(40,40),BETA
      COMMON/NODPNTS/ NOFUEL,INNODES,NODENC,NTHREG
      SUMTOP=0.
      SUMBOT=0.
C
C         SELECT TEMPERATURE OF NEIGHBORING NODE
C
      DO 20 K=1,4
      GO TO(14,15,16,17) ,K
   14 IF(JDUM.EQ.NODENC) GO TO 18
      TEMPNBR=TEMP(IDUM,JDUM+1)
      GO TO 19
   15 IF(IDUM.EQ.1) GO TO 20
      TEMPNBR=TEMP(IDUM-1,JDUM)
      GO TO 19
   16 IF(JDUM.EQ.1) GO TO 20
      TEMPNBR=TEMP(IDUM,JDUM-1)
      GO TO 19
   17 IF(IDUM.EQ.NTHREG) GO TO 20
      TEMPNBR=TEMP(IDUM+1,JDUM)
      GO TO 19
   18 TEMPNBR=TEMPBLK
```

```
 19    CONTINUE
       SUMTOP=SUMTOP+COND(IDUM,JDUM)*AREA(IDUM,JDUM,K)*
      *TEMPNBR/DS(IDUM,JDUM,K)
       SUMBOT=SUMBOT+COND(IDUM,JDUM)*AREA(IDUM,JDUM,K)/
      *DS(IDUM,JDUM,K)
 20    CONTINUE
C
C          CALCULATION OF TEMPERATURE
C
       TEMP(IDUM,JDUM)=(QK(IDUM,JDUM)*VOLUME(IDUM,JDUM)+
      *SUMTOP)/SUMBOT
C
C          OVERRELAXATION OF TEMPERATURE
C
       TEMP(IDUM,JDUM)=BETA*TEMP(IDUM,JDUM)+(1-BETA)*
      *OLDTEMP(IDUM,JDUM)
       TEMPNEW=TEMP(IDUM,JDUM)
       RETURN
       END
```

PROGRAM LISTING FOR MDL-2

```
      PROGRAM MOL2 (INPUT,OUTPUT,TAPE5,TAPE6)
      DIMENSION AREA(40,40,4), BS(40,40,4)
      DIMENSION VOLUME(40,40),TEMP(40,40), OLDTEMP(40,40)
      DIMENSION RADLIN(50),RADNOD(40),THELIN(50),THENOD(40)
      DIMENSION TDIFF(2)
      DIMENSION COND(40,40),QK(40,40)
      COMMON/HEATSRC/SOURCE
      COMMON/TEMPOLD/ OLDTEMP(40,40)
      COMMON/TEMP/ TEMP(40,40),TEMPBLK,AREA(40,40,4)
      COMMON/BETA/ BS(40,40,4),QK(40,40),VOLUME(40,40),BETA
      COMMON/NODPNTS/NOFUEL,INNODES,NODEND,NTHREG
      COMMON/CONDUCT/COND(40,40)
      COMMON/SPACING/RADNOD(40),RADLIN(50),THESPAC
      PI=3.14159265536
C
C        INPUT
C
      READ(5,2000) NTHREG
      READ(5,2000) NOFUEL
      READ(5,2000) NOCLAD
      READ(5,2001) BETA
      READ(5,2001) EPS
      READ(5,2001) CONDCLA
      READ(5,2001) CONDGAS
      READ(5,2001) TEMP1ST
      READ(5,2001) TEMPBLK
      READ(5,2000) NALLOWD
      READ(5,2001) RCLAD
      READ(5,2001) CONDFUE
      READ(5,2001) SHIFT
      READ(5,2001) RFUELA
      READ(5,2001) GASWDTH
      READ(5,2001) SOURCE
 2000 FORMAT(1X,I4)
 2001 FORMAT(1X,F12.8)
C
C          DETERMINATION OF NODAL LINE SPACINGS
C
      INNODES=NOFUEL+1
```

```
      LINEND=NOFUEL+NOCLAD+2
      NODEND=LINEND-1
      THESPAC=PI/NTHREG
      DIFFER=SHIFT*GASWDTH
      RFUELD=RFUELA-DIFFER
      AREAFD=PI*RFUELA*RFUELA/NOFUEL
      RINCLAD=RFUELA+GASWDTH
      AREACA=(RCLAD*RCLAD-RINCLAD*RINCLAD)*PI/NOCLAD
      RADLIN(1)=0.
      DO 1 J=2,LINEND
      IF(J.EQ. INNODES+1) GO TO 2
      IF(J.GT. INNODES+1) GO TO 3
      RADLIN(J)=SQRT((J-1)*AREAFD/PI)
      GO TO 4
    2 RADLIN(J)=RFUELA+GASWDTH
      GO TO 4
    3 RADLIN(J)=SQRT((RFUELA+GASWDTH)**2+(J-INNODES-1)*
     vAREACA/PI)
    4 RADNOD(J-1)=(RADLIN(J)+RADLIN(J-1))/2.
    1 CONTINUE
      THELIN(1)=0.
      DO 5 J=1,NTHREG
      THELIN(J+1)=THELIN(J)+THESPAC
      THENOD(J)=(THELIN(J)+THELIN(J+1))/2.
    5 CONTINUE
C
C          INITIALIZE NODAL TEMPERATURES
C          AND ASSIGN ‘0’ VALUES
C
      DO 50 I=1,NTHREG
      DO 50 J=1,NODEND
      TEMP(I,J)=TEMP1ST
      QK(I,J)=0.
      IF(J.LE.NOFUEL  ) QK(I,J)=SOURCE
C
C          ASSIGN CONDUCTIVITY VALUES
C
      COND(I,J)=CONDFUE
      IF(J.EQ.INNODES) GO TO 6000
```

```
      IF(J.GT.INNODES) COND(I,J)=CONDCLA
      GO TO 50
C
C          MODELED CONDUCTIVITY AS A FUNCTION OF ANGLE
C
 6000 IF(SHIFT.EQ.0.0) GO TO 6001
      PHI=THENOD(I)
      CS=COS(PHI)
      SN=SIN(PHI)
      R=(RFUELA+GASW)TH
      G=R-(DIFFER*CS+SQRT(RFUELA**2-DIFFER**2*(1-CS**2)))
      PHIOTH =ACOS(CS*SQRT(1-(DIFFER/RFUELA)**2*SN**2)-
     v(DIFFER/RFUELA)*SN**2)
      DCOS=COS(PHIOTH-PHI)
      DSIN=SIN(PHICTH-PHI)
      COND(I,J)=CONDGAS*(DCOS+DSIN)*GASWOTH/G
      GO TO 50
 6001 COND(I,J)=CONDGAS
 50   CONTINUE
C
C          CALCULATE NODAL VOLUMES AND SURFACE AREAS
C
      DO 6 I=1,NTHREG
      DO 6 J=1,NODEND
      VOLUME(I,J)=(RADLIN(J+1)**2-RADLIN(J)**2)*THESPAC/2.
      DO 6 K=1,4
      GO TO (7,8,9,10),K
 7    AREA(I,J,K)=RADLIN(J+1)*THESPAC
      GO TO 12
 8    IF(I.EQ.1) GO TO 11
      AREA(I,J,K)=RADLIN(J+1)-RADLIN(J)
      GO TO 12
 9    AREA(I,J,K)=RADLIN(J)*THESPAC
      GO TO 12
 10   IF(I.EQ.NTHREG) GO TO 11
      AREA(I,J,K)=RADLIN(J+1)-RADLIN(J)
      GO TO 12
 11   AREA(I,J,K) =0.
C
```

```
C          CALCULATE INTERNODAL DISTANCES
C
 12    ES(I,J,K)=GETBS(I,J,K)
 6     CONTINUE
C
C          INITIALIZE ITERATION COUNTERS
C
       ITCOUNT=0
 29    NSPEED=0
 23    IF( ITCOUNT.GT. NALLOWD) GO TO 30
       IF(NSPEED.GE. 49) GO TO 25
C
C          ITERATION CALCULATION OF TEMPERATURES
C
 27     DO 21 J=1,NODEND
       DO 21 I=1,NTHREG
       II=NTHREG+1-I
       OLDTEMP(II,J)=TEMP(II,J)
       TEMP(II,J)=TEMPNEW(II,J)
 21    CONTINUE
C
C          CHECK FOR CONVERGENCE
C
 75    DO 24 I=1,NTHREG
       DO 24 J=1,NODEND
       IF(ABS(TEMP(I,J)/OLDTEMP(I,J)-1).GT.EPS) GO TO 22
       GO TO 24
 22    ITCOUNT=ITCOUNT+1
       NSPEED=NSPEED+1
       GO TO 23
 24    CONTINUE
       GO TO 30
C
C          DOMINANT-ERROR-MODE ACCELERATION
C
 25    TDSUM=0.
       DO 26 I=1,NTHREG
       DO 26 J=1,NODEND
       TRATIO=ABS(OLDTEMP(I,J)/TEMP(I,J)-1.)
```

```
      TDSUM=TDSU1+TRATIO
 26   CONTINUE
      TDIFF(NSPEED-48)=TDSUM
      IF(NSPEED.NE.50) GO TO 27
      ARATIO=TDIFF(2)/TDIFF(1)
      ALPHA=0.0
      IF(ARATIO.NE.1.0) ALPHA=1.0/(1.-ARATIO)
      IF(ARATIO.GE.1.0) GO TO 29
      DO 28 I=1,NTHREG
      DO 28 J=1,NODEND
 28   TEMP(I,J)=ALPHA*TEMP(I,J)+(1.-ALPHA)*OLDTEMP(I,J)
      GO TO 29
C
C          OUTPUT
C
 30   WRITE(6,40)RFUELA,RCLAD,GASWDTH,SHIFT,SOURCE,CONDFUE,
     vCONDCLA,CONDGAS,EPS
 40   FORMAT(62X,*INPUT DATA*,//,55X,*RADIUS OF FUEL=*,F10.7
     v,/,55X,*RADIUS OF CLAD=*,F10.7,/,55X,*GAS WIDTH=*,F10.
     v7,/,55X,*FUEL SHIFT=*,F10.7,/,55X,*HEAT GENERATION=*,
     vF12.7,/,55X,*CONDUCTIVITY OF FUEL=*,F10.7,/,55X,
     v*CONDUCTIVITY OF CLAD=*,F10.7,/,55X,
     v*CONDUCTIVITY OF GAS=*,F10.7,/,55X,*EPSILON=*,F10.7,
     v///)
      WRITE(6,41) ITCOUNT
 41   FORMAT(45X,*FINAL TEMPERATURE AFTER*,2X,I4,2X,
     v*ITERATIONS*,/)
      NPREV=1
      NCOUNT=10
 31   WRITE(6,42)
      DO 33 I=1,NTHREG
      II=NTHREG+1-I
      IF(NCOUNT.GT.NODEND)NCOUNT=NODEND
      WRITE(6,32)(TEMP(II,J),J=NPREV,NCOUNT)
 32   FORMAT(5X,**,2X,10(F8.3,4X),2X,**,/)
 33   CONTINUE
      WRITE(6,42)
 42   FORMAT(5X,126(**) )
      WRITE(6,43)(RADNOD(J),J=NPREV,NCOUNT)
```

```
  43    FORMAT(1X,'SPACING',10(F8.3,4X),//)
        NPREV=NCOUNT+1
        NCOUNT=NCOUNT+10
        IF(NCOUNT-NODEND.LT.10) GO TO 31
        END
        FUNCTION GETBS(L,M,N)
C
C           FUNCTION TO DETERMINE INTERNODAL DISTANCES FOR EACH
C           NODE TO EACH OF 4 NEIGHBORS (OR BOUNDARY)
C
        COMMON/CONDUCT/COND(40,40)
        COMMON/NODPNTS/NOFUEL,INNODES,NODEND,NTHREG
        COMMON/SPACING/RADNOD(40),RADLIN(50),THESPAC
        GO TO (1,2,3,4),N
C
C           OUTSIDE FACE OF NODE
C
  1     IF(M.EQ.NOFUEL  ) GO TO 20
        IF(M.EQ.INNODES) GO TO 20
        IF(M.EQ.NODEND) GO TO 30
        GETBS=RADNOD(M+1)-RADNOD(M)
        RETURN
C
C           RIGHT SIDE FACE OF NODE
C
  2     IF(L.EQ.1) GO TO 50
        GETBS=RADNOD(M)*THESPAC
        RETURN
C
C           INSIDE FACE OF NODE
C
  3     IF(M.EQ.1) GO TO 50
        IF(M.EQ.INNODES    ) GO TO 60
        IF(M.EQ.INNODES+1) GO TO 60
        GETBS=RADNOD(M)-RADNOD(M-1)
        RETURN
C
C           LEFT SIDE FACE OF NODE
C
```

```
    4     IF(L.EQ. NTHREG) GO TO 50
          GETBS=RADNOD(M)*THESFAC
          RETURN
   20     GETBS=RADLIN(M+1)-RADNOD(M)+(RADNOD(M+1)-RADLIN(M+1))
         $*COND(L,M)/COND(L,M+1)
          RETURN
   30     GETBS=RADLIN(M+1)-RADNOD(1)
          RETURN
   50     GETBS=1.
          RETURN
   60     GETBS=RADNOD(M)-RADLIN(M)+(RADLIN(M)-RADNOD(M-1))
         $*COND(L,M)/COND(L,M-1)
          RETURN
          END
          FUNCTION TEMPNEW(IDUM,JDUM)
C
C              FUNCTION TO CALCULATE NODAL TEMPERATURES
C
          COMMON/TEMP/ TEMP(40,40),TEMPBLK,AREA(40,40,4)
          COMMON/CONDUCT/ COND(40,40)
          COMMON/TEMPOLD/ OLDTEMP(40,40)
          COMMON/BETA/ BS(40,40,4),QK(40,40),VOLUME(40,40),BETA
          COMMON/NODPNTS/ NOFUEL,INNODES,NODEND,NTHREG
          SUMTOP=0.
          SUMBOT=0.
C
C              SELECT TEMPERATURE OF NEIGHBORING NODE
C
          DO 20 K=1,4
          GO TO(14,15,16,17) ,K
   14     IF(JDUM.EQ.NODEND) GO TO 18
          TEMPNBR=TEMP(IDUM,JDUM+1)
          GO TO 19
   15     IF(IDUM.EQ.1) GO TO 20
          TEMPNBR=TEMP(IDUM-1,JDUM)
          GO TO 19
   16     IF(JDUM.EQ.1) GO TO 20
          TEMPNBR=TEMP(IDUM,JDUM-1)
          GO TO 19
```

```
 17    IF(IDUM.EQ.NTHREG) GO TO 20
       TEMPNBR=TEMP(IDUM+1,JDUM)
       GO TO 19
 18    TEMPNBR=TEMPBLK
 19    CONTINUE
       SUMTOP=SUMTOP+COND(IDUM,JDUM)*AREA(IDUM,JDUM,K)*
      &TEMPNBR/DS(IDUM,JDUM,K)
       SUMBOT=SUMBOT+COND(IDUM,JDUM)*AREA(IDUM,JDUM,K)/
      &DS(IDUM,JDUM,K)
 20    CONTINUE
C
C          CALCULATION OF TEMPERATURE
C
       TEMP(IDUM,JDUM)=(QK(IDUM,JDUM)*VOLUME(IDUM,JDUM)+
      &SUMTOP)/SUMBOT
C
C          OVERRELAXATION OF TEMPERATURE
C
       TEMP(IDUM,JDUM)=BETA*TEMP(IDUM,JDUM)+(1-BETA)*
      &OLDTEMP(IDUM,JDUM)
       TEMPNEW=TEMP(IDUM,JDUM)
       RETURN
       END
```

B.2 RFIND

RFIND is a simple geometrical program developed to translate the points within the ratchet boundary coordinate system, (ρ,ϕ), into the corresponding coordinate system for the modeled conductivity systems, (r,θ). The translation of points is necessary to obtain a direct comparison of the temperatures predicted by the different models.

Equations 2.1 and 2.2 from Chapter II are used to generate the following radial and angular relationships between the two coordinate systems:

$$R = [\rho^2 + d^2 - 2\,\rho d\,\cos(\phi)]^{\frac{1}{2}} \qquad \text{B-1}$$

$$\theta = \tan^{-1}[\sin(\phi)/(\cos(\phi) - d/\rho)] \qquad \text{B-2}$$

RFIND uses the same input as required by the other programs. A listing of RFIND is presented below.

PROGRAM LISTING FOR RFIND

```
      PROGRAM RFIND (INPUT,OUTPUT,TAPE5,TAPE6)
      DIMENSION RADLIN(50), RADNOD(40), THELIN(50),
     *THENOD(40)
      DIMENSION R(40,40),PHIOTH(40,40)
      PI=3.14159265536
C
C          INPUT
C
      READ(5,2000) NTHREG
      READ(5,2000) NOFUEL
      READ(5,2000) NOCLAD
      READ(5,2001) BETA
      READ(5,2001) EPS
      READ(5,2001) CONDCLA
      READ(5,2001) CONDGAS
      READ(5,2001) TEMPIST
      READ(5,2001) TEMPBLK
      READ(5,2000) NALLOWD
      READ(5,2001) RCLAD
      READ(5,2001) CONDFUE
      READ(5,2001) SHIFT
      READ(5,2001) RFUELA
      READ(5,2001) GASWDTH
      READ(5,2001) SOURCE
 2000 FORMAT(1X,I4)
 2001 FORMAT(1X,F12.8)
      INNODES=NOFUEL+NTHREG+1
      LINEND=NOFUEL+NOCLAD+NTHREG+2
      NODEND=NOFUEL+NOCLAD+NTHREG+1
      THESPAC=PI/NTHREG
      DIFFER=SHIFT*GASWDTH
      RFUELD=RFUELA-DIFFER
      AREAFD=PI*RFUELD*RFUELD/NOFUEL
      RINCLAD=RFUELA+GASWDTH
      AREACA=(RCLAD*RCLAD-RINCLAD*RINCLAD)*PI/NOCLAD
      RADLIN(1)=0.
C
C          RADIAL NODE SPACINGS
C
```

```
      DO 1 J=2,LINEND
      IF(J.GT.NOFUEL+1.AND.J.LE.NOFUEL+NTHREG+1) GO TO 2
      IF(J.GT.INNODES) GO TO 3
      RADLIN(J)=SQRT((J-1)*AREAFO/PI)
      GO TO 4
    2 ANGLE=(J-NOFUEL-1)*THESPAC-THESPAC/2.
      RADLIN(J)=(SQRT((2*DIFFER*COS(ANGLE))**2+4*(RFUELA*
     *RFUELA-DIFFER*DIFFER))-2*DIFFER*COS(ANGLE))/2.
      GO TO 4
    3 RADLIN(J)=SQRT(RINCLAD**2+(J-INNODES-1)*AREACA/PI)
    4 RADNOD(J-1)=(RADLIN(J)+RADLIN(J-1))/2.
    1 CONTINUE
      THELIN(1)=0.
C
C         ANGULAR NODE SPACINGS
C
      DO 5 J=1,NTHREG
      THELIN(J+1)=THELIN(J)+THESPAC
      THENOD(J)=(THELIN(J)+THELIN(J+1))/2.
    5 CONTINUE
C
C         RADIAL AND ANGULAR COORDINATE TRANSLATIONS
C
      DO 10 I=1,NTHREG
      DO 10 J=1,NODEND
      CS=COS(THENOD(I))
      SN=SIN(THENOD(I))
      PHIOTH(I,J)=ATAN(SN/(CS-DIFFER/RADNOD(J)))
      R(I,J)=SQRT(RADNOD(J)**2+DIFFER**2-2*RADNOD(J)*
     *DIFFER*CS)
   10 CONTINUE
C
C         OUTPUT
C
      NPREV=1
      NCOUNT=10
   31 DO 33 I=1,NTHREG
      IF(NCOUNT.GT.NODEND) NCOUNT=NODEND
      WRITE(6,32)(R(I,J), J=NPREV,NCOUNT)
```

```
      WRITE(6,32)(PHIOTH(I,J), J=NPREV,NCOUNT)
32    FORMAT(5X,10(F8.5,4X))
33    CONTINUE
      NPREV=NCOUNT+1
      NCOUNT=NCOUNT+10
      IF(NCOUNT-NODEND.LT.10) GO TO 31
      END
```